HP-UX 参考手册

第 1 节：用户命令
(A~M)

HP-UX 11i v2 2004 年 9 月

第 1 卷 （ 共 10 卷 ）



生产部件号： B2355-90931
E0904

法律声明

本文档中的信息如有更改，恕不另行通知。

担保

随 HP 产品及服务提供的明示性担保声明中列出了适用于此 HP 产品及服务的专用担保条款。本文中的任何内容均不构成额外的担保。 HP 对本文中的技术或编辑错误以及缺漏不负任何责任。

美国政府许可

机密计算机软件。必须有 HP 授予的有效许可证，方可拥有、使用或复制本软件。根据供应商的标准商业许可证的规定，美国政府应遵守 FAR 12.211 和 12.212 中有关“商业计算机软件”、“计算机软件文档”与“商业货物技术数据”条款的规定。

附加版权声明

本文档及其所涉及的软件可能同时受到下述一项或多项版权的保护。某些单独的联机帮助页将对这些附加版权加以认可。

商标声明

Intel 和 Itanium 均为 Intel Corporation 在美国和其他国家（地区）的注册商标，使用时已受到许可。

Java 是 Sun Microsystems, Inc. 在美国的商标。

MS-DOS 和 Microsoft 是 Microsoft Corporation 在美国的注册商标。

OSF/Motif 是 The Open Group 在美国和其他国家 （地区）的商标。

UNIX 是 Open Group 的注册商标。

X Window System 是 X/Open Group 的商标。

版本说明

本文档的印刷日期和部件号指明其当前版本。印刷新版本时印刷日期会随之改变。再次印刷时可能会稍作改动，但不会更改印刷日期。新版文档将汇总从上一版本到现在所更新的所有资料。

部件号	日期；发行版；文档格式；发布形式
B2355-90931~40	2004 年 9 月； HP-UX 11i v2 ； PDF （10 卷）； docs.hp.com 和印刷版本。中文版。
B2355-60105	2004 年 9 月； HP-UX 11i v2 ； HTML （1 卷）； docs.hp.com 和 Instant Information。英文版。
B2355-90839~48	2004 年 9 月； HP-UX 11i v2 ； PDF （10 卷）； docs.hp.com 和印刷版本。英文版。
B2355-60103	2003 年 8 月； HP-UX 11i v2 ； HTML （1 卷）； docs.hp.com 和 Instant Information。英文版。
B2355-90779~87	2003 年 8 月； HP-UX 11i v2 ； PDF （9 卷）； docs.hp.com 和印刷版本。英文版。
B9106-90010	2002 年 6 月； HP-UX 11i v1.6 ； HTML （1 卷）； docs.hp.com 和 Instant Information。英文版。
B9106-90007	2001 年 6 月； HP-UX 11i v1.5 ； HTML （7 卷）； docs.hp.com 和 Instant Information。英文版。
B2355-90688	2000 年 12 月； HP-UX 11i v1 ； 9 卷。英文版。
B2355-90166	1997 年 10 月； HP-UX 11.0 ； 5 卷。英文版。
B2355-90128	1996 年 7 月； HP-UX 10.20 ； 5 卷；只有联机版本。英文版。
B2355-90052	1995 年 7 月； HP-UX 10.0 ； 4 卷。英文版。

印刷字体约定

本手册使用下列印刷字体约定。

audit (5)	表示 HP-UX 联机帮助页。“audit”是联机帮助页名称，“5”是该联机帮助页在 《HP-UX 参考手册》中的小节号。在网站和 Instant Information DVD 上，可能是指向该联机帮助页的热链接。在 HP-UX 命令行输入 “man audit” 或 “man 5 audit”可以查看该联机帮助页。详见 *man* (1)。
系统字体	表示计算机显示的文本和系统项 (ComputerOutput)。
强调内容	第一次定义的名词和强调的内容用**黑体**表示。
键盘操作	键盘键名称。注意， Return 和 Enter 指的是同一个键。
《书名》	表示文档中引用的书籍、手册的名称，以宋体表示。
“术语”	表示文档中引用的专用术语，以宋体表示。
[ ]	格式和命令说明中的可选内容。如果内容用 “|”分隔，就必须选择其中之一。
{ }	格式和命令说明中必需的内容。如果内容用 “|”分隔，就必须选择其中之一。
...	前面的元素可以重复任意多次。
|	分隔选项列表中的项目。

前言

HP-UX 是 Hewlett-Packard Company 开发的一种操作系统，可与各种行业标准兼容。该操作系统基于 UNIX® System V Release 4 操作系统，并包括 Fourth Berkeley Software Distribution 中的重要功能。

本手册包括系统参考文档资料，即联机帮助页。单个的文档也称为联机帮助页和参考页。

简介

有关 HP-UX 以及联机帮助页的结构和格式的简介信息，请参阅 *introduction* (9) 联机帮助页。

小节简介

联机帮助页分为若干节，各节也包含 introduction 或 intro 联机帮助页，该联机帮助页介绍具体的内容。这些节有：

intro (1)	第1节：用户命令
intro (1M)	第1M节：系统管理命令
intro (2)	第2节：系统调用
intro (3C)	第3节：库函数
intro (4)	第4节：文件格式
intro (5)	第5节：其他主题
intro (7)	第7节：设备专用文件
intro (9)	第9节：简介和词汇表

第 1 卷

目录

第 1 节

第 1 卷

目录

第 1 节

目录
第 1 、2 卷

第 1 节：用户命令

条目名称(节编号)：名称 **说明**

目录
第 1 、2 卷

目录
第 1 、2 卷

目录
第 1 、2 卷

条目名称(节编号)：名称 **说明**

条目名称(节编号)：名称 说明

目录
第 1 、2 卷

条目名称(节编号)：名称 说明

目录
第 1 、2 卷

目录
第 1 、2 卷

条目名称(节编号)：名称 说明

第 1 节
第 1 部分

用户命令
A~M

第 1 节
第 1 部分

用户命令
A~M

名称

intro - 命令实用程序和应用程序简介

说明

本节介绍了可由用户访问的命令，而第 (2) 节中的系统调用或第 (3) 节中的库例程则可由用户程序访问。

命令语法

除非另行说明，否则本节介绍的命令将按照以下语法使用选项和其他参数：

name [*option* (*s*)] [*cmd_arg* (*s*)]

其中的元素定义如下：

name 可执行文件的名称。

option 命令行中可以出现一个或多个 *option* 。每一个均采用下列形式之一：

-*no_arg_letter*
单个字母，表示不带参数的选项。

-*no_arg_letters*
两个或更多单字母选项组合为一个命令行参数。

-*arg_letter***<>***opt_arg*
一个单字母选项后跟所需参数，其中：

arg_letter
是单个字母，它表示需要参数的选项。

opt_arg
是一个参数（字符串），它满足前面的 *arg_letter* 。

<> 表示可选空格。

cmd_arg 不以 **-** 开头的路径名（或其他命令参数），或者是 **-** 本身，此时它表示标准输入。如果出现两个或更多 *cmd_arg* ，则必须用空格分隔它们。

手册条目格式

所有手册条目都遵循已制定的主题格式，但并不是每个条目中都包含所有主题。

名称 给出条目的名称，并简要陈述其用途。

概要 概述所介绍的条目或程序实体的用法。使用下列约定：

Computer font 字符串是系统文本，应该完全按照它们在手册中显示的形式进行键入（第 2 节和第 3 节中条目的“概要”部分中的参数除外）。

Italic 字符串表示可替换的参数名以及可在手册其他部分中找到的手册条目名。

用方括号 [] 括起的参数名表示该参数为可选参数。

省略号 (...) 用于表示先前的参数可重复。

命令自身使用最终约定。以短划线 (-)、加号 (+) 或等号 (=) 开头的参数，通常被视为某种选项参数，即使它出现在文件名可能出现的位置上也是如此。因此，不要让文件名以 -、+ 或 = 字符开头。

说明　讨论每个条目的功能和行为。

外部语言环境影响　该标题下的信息涉及到各地语言的程序设计。典型的条目是，对单字节和（或）多字节字符的支持、与语言相关的环境变量对系统行为的影响，以及其他相关信息。

网络功能　仅当使用这一部分介绍的网络功能（如 NFS）时，该标题下的信息才适用。

返回值　讨论在完成程序调用时返回的各个值。

诊断信息　讨论可能生成的诊断信息。但不列出自述性消息。

错误　列出错误条件及其相应的错误消息或返回值。

举例　在适当的地方提供典型用法的示例。

警告　指出可能存在的缺陷。

相关内容　指出 HP-UX 操作中与用户或者特定硬件或硬件组合相关的变化形式。

作者　指明由此手册条目记录的软件的来源。

文件　列出组成程序或命令的文件名。

另请参阅　提供相关主题的指示性信息。

缺陷　讨论已知问题与缺陷，有时提供建议的解决方法。

符合的标准　这一部分列出 HP-UX 组件所符合的标准规范。

返回值

终止时，每个命令都返回两个状态字节，一个由系统提供，它给出终止的原因；另一个（在“正常”终止的情况下）由程序提供（有关说明，请参阅 *wait*(2) 和 *exit*(2)）。在正常终止时，系统提供的字节是 0。程序提供的字节通常是 0 （表示成功执行）和非零值（表示错误或失败，例如命令行中的参数不正确、数据无效或数据不可访问）。返回的值通常有不同的称谓：“退出代码”、“退出状态”、“返回代码”或“返回值”，并且仅在涉及特殊约定时进行说明。

警告

当所处理的文件包含空字符时，某些命令会产生不可预料的结果。这些命令通常将文本输入行视为字符串，因此，在一行内遇到空字符（字符串结束符）时会产生混淆。

另请参阅

getopt(1)、 exit(2)、 wait(2)、 getopt(3C)、 hier(5)、 introduction(9)。

可以通过 Web 访问 HP-UX 文档，网址为： **http://docs.hp.com** 。

名称

adb - 绝对调试器

概要

adb [**-h**]

adb [**-n**|**-o**] [**-w**] [**-I** *path*] *kernelfile memfile*

adb [**-n**|**-o**] [**-w**] [**-I** *path*] *kernelfile crashdir*

adb [**-n**|**-o**] [**-w**] [**-I** *path*] *crashdir*

adb [**-n**|**-o**] [**-w**] [**-I** *path*] [*objfile*] [*corefile*]

adb [**-n**|**-o**] [**-w**] [**-I** *path*] **-P** *pid* [*execfile*]

说明

adb 命令执行一种通用的调试程序，该调试程序对运行它的处理器和操作系统的基础架构很敏感。它可用于检查文件并为执行 HP-UX 程序提供一个可控制的环境。

adb 会确切地检查一个对象文件（称为 "当前对象文件"）和一个内存文件（称为 "当前内存文件"）。这两个文件中的任一个文件都可以是 NULL（空）文件（通过使用 **-** 参数进行指定），即没有任何内容的文件。可使用下列参数指定对象文件和内存文件：

kernelfile HP-UX 内核，通常为 **vmunix** 。

memfile **/dev/mem** 或 **/dev/kmem** 。如果已指定 *kernelfile* ，则假定 *memfile* 位于运行 *kernelfile* 的 HP-UX 系统上。仅在 PA-RISC 平台上才支持 **/dev/mem** 。

crashdir 包含 HP-UX 系统故障转储的目录，如果已指定 *kernelfile* ，则假定该目录由 *kernelfile* 生成。

objfile 通常是一个可执行程序文件。它也可以是浮动对象文件、共享库文件或 DLKM 模块。 *objfile* 的缺省值是 **a.out** 。

corefile 执行 *objfile* 后生成的核心映像文件。 *corefile* 的缺省值是 **core** 。

execfile 与 *pid* （即 **adb** 进行调试要采用的进程的进程 ID）对应的可执行文件。

当前对象文件可以是下列文件之一： *kernelfile* 、 *crashdir* 中的 **vmunix** 文件、 *objfile* 或 *execfile* 。当前对象文件最好应包含一个符号表；如果没有包含，那么虽然仍可以检查该文件，但无法使用 **adb** 的符号功能。当前内存文件可以是下列文件之一： *memfile* 、 *crashdir* 中的系统内存转储、 *corefile* 或 *pid* 的内存。

对 **adb** 的请求从标准输入中读取， **adb** 在标准输出上作出响应。如果存在 **-w** 标志，则将创建（如有必要）并打开 *objfile* 以便进行读取和写入，随后使用 **adb** 可以对该文件进行修改。 **adb** 忽略 QUIT；使用 INTERRUPT 可返回至下一个 **adb** 命令。

adb 有两种操作模式：向后兼容模式和普通模式。向后兼容模式是 PA-RISC 系统上的缺省模式。普通模式是 Itanium 系统上的缺省模式。

启动时， **adb** 从文件 **$HOME/.adbrc** 中执行 **adb** 命令。

要调试 MxN 进程或核心， **adb** 需要 MxN 调试库 **libmxndbg** 。根据应用程序的类型，它会加载 **/usr/lib/libmxndbg.sl** （对于 32 位 PA-RISC 系统）、 **/usr/lib/libmxndbg64.sl** （对于 64 位 PA-RISC 系统）或 **/usr/lib/hpux32/libmxndbg.so** （对于基于 Itanium(R) 的系统）。如果没有在指定路径中找到相关的库，则应将 Shell 变量 **ADB_PATHMXNDBG** 设置为可以找到正确库的路径。

选项

adb 可识别下列命令行 选项，这些选项可以按任意顺序出现，但必须位于所有文件参数之前：

-h 输出用法摘要并退出。如果使用此选项，则忽略其他选项和参数。

-i 忽略 **$HOME/.adbrc** 。

-I *path* *path* 指定一个目录列表，在这些目录中搜索使用 **<** 或 **<<** （请参阅下文）读取的文件。此列表使用与 **PATH** Shell 变量相同的语法以及与之相似的语义；缺省值为 **.:/usr/lib/adb** 。

-n 指定普通模式。此模式是 Itanium 系统上的缺省模式。此选项与 **-o** 选项相互排斥。最后指定的那个选项有效。

-o 指定向后兼容模式。此模式为 PA-RISC 系统上的缺省模式。此选项与 **-n** 选项相互排斥。最后指定的那个选项有效。

-P *pid* 将进程 ID 为 *pid* 的进程用作“受跟踪”的进程；请参阅 *ttrace*(2) 。此选项对于调试最初不是在 **adb** 控制下运行的进程很有帮助。

-w 必须指定此选项以启用 **adb** 的文件写入命令。将打开 *Objfile* 以便进行读取和写入。如果对象文件是内核内存文件，则此选项还会启用对 *memfile* 的写入。

adb 的下列命令行选项已过时并且不再需要。（如果使用它们，则会生成警告）

-k 以前， **adb** 需要使用此选项来识别 HP-UX 故障转储或 **/dev/mem** 。

-m 以前， **adb** 需要使用此选项来识别多个文件的 HP-UX 故障转储。

对 **adb** 的请求可采用传统格式：

[*address*] [**,***count*] [*command-char*] [*command-arguments*] [**;**]

或新格式：

keyword [*command-arguments*] [**;**]

在向后兼容模式中，只能使用传统格式。

如果存在 *address* ，则将 **dot** 设置为 *address* 。 **dot** 是跟踪当前地址的 **adb** 状态变量。 **dotincr** 是另一个状态变量，它在 **adb** 逐步执行格式字符串时跟踪 **dot** 的增量；请参阅下文的格式字符串。最初， **dot** 和 **dotincr** 均设置为 **0** 。对于大多数命令， *count* 指定命令要执行的次数。缺省 *count* 为 **1** 。 *address* 和 *count* 都是表达式。

对地址的解释取决于使用地址的上下文。如果在调试子进程，则在该子进程的地址空间内解释地址。（有关详细信息，请参阅下文的地址映射）

command-char 和 *command-arguments* 指定要运行的命令。请参阅下文的命令。

表达式

所有的 **adb** 表达式基元都被视为 64 位无符号整数，表达式也按 64 位无符号整数进行求值。表达式支持下列基元：

integer 一个数值。前缀 **0** （零）、 **0o** 和 **0O** 强制以八进制基数进行解释；前缀 **0t** 、 **OT** 、 **0d** 和 **0D** 强制以十进制基数进行解释；前缀 **0x** 和 **0X** 强制以十六进制基数进行解释；前缀 **0b** 和 **0B** 强制以二进制基数进行解释。因此， **020 = 0d16 = 0x10 = 0b1000 =** 16。如果不带前缀，则使用缺省基数；请参阅 **d** 命令。基数初始化时设置为十六进制。请注意，最高有效位可能为字母字符的十六进制数必须带 **0x** （或 **0X** ）前缀。

'*cccccccc***'** 最多为 8 个字符的 ASCII 值。如果指定了多于 8 个字符的值，则该值是未定义的。反斜杠 (\) 可用于对单引号 (**'**) 进行转义。

$*register* 寄存器。寄存器的值从当前内存文件所对应的寄存器组中获得。寄存器名称与系统实现相关；请参阅 **r** 命令。

symbol *symbol* 是大小写字母、下划线或数字组成的序列，但不能以数字开头。反斜杠 (\) 可用于对其他字符进行转义。 *symbol* 的值从当前对象文件的符号表中获得。

variable 变量名由字母和数字组成，并始终以 $ 开头。目标处理器中寄存器的名称保留为变量名，并可用于在表达式中访问寄存器。

在向后兼容模式中，除寄存器之外，变量均是单个数字或字母，并且前缀字母是 **>** 。

下面列出了两种模式中都支持的变量。

9 最后一个 **$<** 命令上的计数。

b 数据段的基址。

d 数据段大小。

t 文本段大小。

e 入口点。

s 堆栈段大小。

m “幻数”，如 **<magic.h>** 中所定义。

输入时， *b* 、 *d* 和 *t* 通过当前内存文件中的标题进行设置。如果当前内存文件似乎无效，则通过当前对象文件设置这些值。 *e* 通过当前对象文件进行设置。

注释：上述变量只能通过核心文件和对象文件进行设置。

下列基元仅在普通模式中才受支持：

$. **dot** 的值。

$+ **dot** 的值，以 **dotincr** 的值为增量增加。

$- **dot** 的值，以 **dotincr** 的值为减量减少。

$~ 键入的最后一个 *address* 。

下列基元仅在向后兼容模式中才受支持：

. **dot** 的值。

+ **dot** 的值，以 **dotincr** 的值为增量增加。

^ **dot** 的值，以 **dotincr** 的值为减量减少。

" 键入的最后一个 *address* 。

支持以下 C 算术运算符、关系运算符和逻辑运算符，并且其优先级与在 C 中相同：

?: || && | ^ & == != < > <= >= >> << + - * / % ~ !

还支持 C 的一元符号运算符 **+** 和 **-** 以及 () 运算符。

除上述运算符之外，还支持下列 **adb** 特定的一元运算符，并且其优先级与其他一元运算符相同：

*****_exp_ 在当前内存文件中，由 *exp* 指明的位置中的内容。

@*exp* 在当前对象文件中，由 *exp* 指明的位置中的内容。

以下 adb 特定的二元运算符具有与 **%** 运算符相同的优先级：

*exp1***#***exp2* *exp1* 向上舍入到最接近于 *exp2* 的倍数的值。

在向后兼容模式中， **%** 运算符的语义与 **/** 运算符相同。一元 **+** 运算符在此模式中不可用。

如果包含任何以下运算符的子表达式用在 *address* 或 *count* 表达式中，则应该使用 () 将子表达式保护起来：

?: / $!

除上述运算符之外，在向后兼容模式中还应保护 **>** 运算符。

命令

如上所述， **adb** 命令可以使用 “传统格式” 或 “关键字格式” 进行指定。在向后兼容模式中，仅支持传统格式。

“传统格式命令”

下列类别的命令将采用传统命令格式指定：

- 文件命令
- 关键字命令
- 进程命令

- 线程命令
- Shell 命令

在向后兼容模式中：

- 变量命令

文件命令

下列命令作用于当前对象文件或当前内存文件，可用于读取、写入等。

file_selector[*modifier*] [**,***size* | *index*] [*arglist*]

file_selector 可以是下列各项之一：

? 所选文件是当前对象文件。

/ 所选文件是当前内存文件。

= 此特殊符号仅用于输出 **dot** 的值。

modifier 指定对文件的操作； *modifier* 可以是：

(no modifier) 使用单个可选参数列表，该列表为格式字符串。 **adb** 根据该格式字符串从所选文件中输出数据。如果格式字符串不存在，并且文件选择器是 **?** 或 **/** ，则 **adb** 使用该文件选择器以前使用的格式字符串。如果文件选择器是 **=** ，并且格式字符串不存在，则 **adb** 使用前一个 **=** 命令使用的格式字符串。

/ [**,** *size*] *value* [*mask*]

搜索所选文件。就大小而言，以 **dot** 开头的 *size* 使用 *mask* 进行屏蔽，并与 *value* 进行比较，直到找到一个匹配项。如果找到匹配项，则将 **dot** 设置为被屏蔽对象的地址。如果省略掩码，则不使用任何掩码。 **dotincr** 设置为 0。大小的有效值是 1、2、4、8。如果未指定 *size* ，则假定为 **sizeof(int)** 。 *value* 和 *mask* 都是 *size* 字节的无符号整数。

例如： *expr***?/,4 6 5** 。搜索 4 字节值，在当前对象文件中写入以 *expr* 开头的 4 (6 & 5)。

= [**,** *size*] *value1 value2 ...*

在寻址位置写入一个指定了大小的 *size* 值。每次写入后， **dot** 以 *size* 为增量增加。 **dotincr** 设置为 0。 *size* 和 *values* 与 **/** 修饰符中的对应内容相同。对于此操作，应该使用 **-w** 选项打开文件。

例如： *expr***?=,4 6 5** 。在当前对象文件中，分别在地址 *exp* 和 *exp*+4 处写入 以exp 开头的 6 & 5。

> [**,** *index*] *b e f*

将所选文件的第 *index* 个映射三元组参数按顺序设置为相应的参数值。请参阅地址映射。如果给出的参数少于三个，则剩余的映射保持不变。这些参数为表达式。如果未指定，则假定 *index* 为 0。例如： **?>,0 1 2 3** 将当前对象文件的 *b* 、 *e* 、 *f* （索引 0）分别设置为 1、2、3。

在向后兼容模式中，还提供了下列修饰词。

- ***** 其行为与无修饰词时相同。但是，它使用第二个映射三元组查找要输出的数据的文件地址。
- **l** 其行为与带隐含大小 2 的修饰词 **/** 相同。它会将 **dotincr** 设置为 2。
- **L** 其行为与带隐含大小 4 的修饰词 **/** 相同。它会将 **dotincr** 设置为 4。
- **w** 其行为与带隐含大小 2 的修饰词 **=** 相同。它会将 **dotincr** 设置为 2。此修饰词使 **dot** 增加，增量为所有写入值的总大小减去 **dotincr** 。
- **W** 其行为与带隐含大小 4 的修饰词 **/** 相同。它会将 **dotincr** 设置为 4。 **dot** 的设置与 **w** 中一样。
- **m** 其行为与带隐含 *index* 0 的修饰词 **>** 相同。
- ***m** 其行为与带隐含 *index* 1 的修饰词 **>** 相同。

对于这些修饰词，不能指定任何显式的 *size* 或 *index* 。不推荐使用这些修饰词。

关键字命令

通过在命令前加前缀 **$** ，可使用传统命令格式运行关键字命令格式。有关关键字命令的完整列表，请参阅关键字格式命令。

进程命令

这些命令用于处理所管理的子进程。 **adb** 可运行一个对象文件作为子进程。此外，如果给定了子进程的 pid，它还可以采用子进程。 **adb** 可以调试多线程的子进程和（或）派生的子进程。它还可以同时调试多个子进程。但是，在任何时候它都集中处理一个子进程和其中一个线程，它们分别称为 “当前子进程”和 “当前线程”。

进程命令由 **:** 后跟 *modifier* 和可选的参数列表组成。它们是：

r [*objfile*] 将 *objfile* 作为子进程运行。如果确切地给定了 *address* ，则从此入口点进入程序；否则，将从标准入口点进入程序。值 *count* 指定在停止之前可忽略的断点数。可以在输入命令的同一命令行上提供 *subprocess* 的 *arguments* 。分号不用作命令分隔符。以 < 或 > 开头的 *argument* 会为命令创建标准输入或输出。进入子进程后，所有信号都开启。这样的子进程就称为 “已创建子进程” 。

如果还有其他已创建子进程正在运行，则会终止所有子进程。它不会终止任何已连接的子进程。这个子进程就成为当前子进程。

e [*objfile*] 与在 **:r** 中一样设置子进程；不执行任何指令。

a [*objfile*] 使 **adb** 采用进程 ID 为 *pid* 的进程作为受跟踪的子进程。如果指定了 *objfile* ，则 **adb** 使用它查找符号信息。计数的含义与 **:r** 中相同。这样的子进程就称为 “已连接子进程” 。这个子进程就成为当前子进程。

k [*pid* | *****] 终止已创建子进程。如果未指定任何参数，则终止当前子进程。如果给定了 *pid* ，则终止具备该 *pid* 的子进程。如果给定了 ***** ，则终止所有已创建子进程。

将从剩余的子进程中选择当前子进程。

de [*pid* | *****] 参数可以是 *pid* 或 ***** 。与 **:k** 相同，但它应用于已连接子进程。 **adb** 可以与这些子进程分离。

c [*signal*] 使用 *signal* 继续执行子进程。它会继续执行该子进程的所有线程。如果未指定任何 *signal*，则发送曾使子进程停止的信号。如果指定了 *address* ，则会从此地址继续执行当前线程。断点跳过与 **:r** 中相同。

s [*signal* | *arg1 arg2 ...*]

步进式执行当前线程 *count* 次。如果给定了 *address* ，则当前线程从该 *address* 继续执行，其他线程则从停止处的 *address* 继续执行。如果未指定任何 *signal* ，则发送曾使线程停止的 *signal* 。如果无当前子进程，则运行对象文件作为子进程，与 **:r** 中相同。这种情况下，不能发送信号；命令行的剩余部分视为该子进程的参数。

b [*command*]

在当前子进程中的 *address* 处设置断点。将执行断点 *count*-1 次，然后停止。每次遇到断点时，就执行 *command* 。此断点是一个 *subprocess* 断点。如果任一线程在此 *address* 执行指令，则该线程将停止。同一 *address* 上可以设置多个断点。

d [*num* | *****] 如果指定了此选项，则删除当前子进程中 *address* 处的所有断点。如果指定了 *****，则删除所有当前子进程断点。如果指定了 *num* ，则删除编号为 *num* 的断点。

en [*num* | *****] 如果指定了此选项，则启用当前子进程中 *address* 处的所有断点。如果指定了 *****，则启用所有当前子进程断点。如果指定了 *num* ，则启用编号为 *num* 的断点。

di [*num* | *****] 如果指定了此选项，则禁用当前子进程中 *address* 处的所有断点。如果指定了 *****，则禁用所有当前子进程断点。如果指定了 *num* ，则禁用编号为 *num* 的断点。

z *signum* [**+s** | **-s** | **+r** | **-r** | **+d** | **-d**]

为当前子进程的所有线程更改指定 *signum* 的信号处理。可指定的布置方式如下：

+s 收到 *signum* 时停止 *subprocess* 。

-s 收到 *signum* 时不停止 *subprocess* 。

+r 收到 *signum* 时报告。

-r 收到 *signum* 时不报告。

+d 将 *signum* 发送给目标 *subprocess* 。

-d 不将 *signal* 发送给目标 *subprocess* 。

w [*pid*] 从当前子进程切换到进程 ID 为 *pid* 的 *subprocess* 。此进程就成为当前子进程。此子进程必须是已连接或已创建的子进程。执行此命令之后，两种子进程都处于停止状态。

wc [*pid*] 与 **w** 相同，但不停止前一个当前子进程。

线程命令

这些命令用于管理当前子进程中的线程。线程命令由 **]** 后跟 *modifier* 和可选参数列表组成。

s [*signum*] 与 **:s** 相同。但是，仅限当前线程。

c [*signum*] 与 **:c** 相同。但是，仅继续执行当前线程。 *count* 指针对当前线程要跳过的断点。

b [*command*] 与 **:b** 相同。但是，仅应用于当前线程。

d [*num* | *****] 与 **:d** 相同。但是，仅应用于当前线程。

en [*num* | *****] 与 **:en** 相同。但是，仅应用于当前线程。

di [*num* | *****] 与 **:di** 相同。但是，仅应用于当前线程。

z *signum* [**+s** | **-s** | **+r** | **-r** | **+d** | **-d**]
与 **:z** 相同。但是，仅用于当前线程。如果在此线程的环境中出现 *signum* ，则使用此配置值而不是子进程的配置值。

es [*signum*] 为当前线程设置此 *signum* 的标志。这意味着，如果在此线程信号的环境中出现此 *signum* ，则使用此配置值，而不使用子进程的配置值。

w [*pid*] 从当前线程切换到某些其他线程。切换后两个线程都将处于停止状态，而具备 *threadid* 的线程将成为当前线程。此命令也适用于核心文件调试。它可从当前线程切换到给定线程，并使给定线程成为当前进程。

Shell 命令

此操作由 **!** 字符后跟 *string* 组成。 *string* 将毫无改变地传递给由 **SHELL** 环境变量定义的 Shell 或传递给 **/bin/sh**。

变量命令

仅在向后兼容模式中才支持这种命令。变量命令由 **>** 后跟一个 *variable* 、 var 和一个可选 *value* 组成。此操作会为由 *var* 命名的变量或寄存器赋值。

如果未指定值，则假定 *value* 为 **dot** 的值。不推荐采用此行为。

关键字格式命令

采用此格式的所有命令都由一个关键字后跟一个数目可变的参数组成。

在向后兼容模式中，这些关键字格式命令前面必须有 **$** 。

< *filename* 从 *filename* 中读取命令。如果在文件中执行此命令，则将看不到文件中的其他命令。

在向后兼容模式中，如果给定了 *count* ，则在执行文件中的第一个 *command* 之前，该选项将置于变量 9 中。不推荐采用此行为。

<< *filename* 与 **<** 类似，但是它用于命令文件，且不会导致文件关闭。

在向后兼容模式中，当命令执行时会保存变量 9，当命令完成时会恢复变量 9。不推荐采用此行为。

> *filename* 它会将输出发送到 *filename* ，如果文件不存在，则创建该文件。

在向后兼容模式中，将输出追加到 *filename* 。

>> *filename* 与 **>** 类似，但是将输出追加到 *filename* 。

rp 输出进程 ID 和寄存器值。

r 输出通用寄存器以及进程计数器寻址的指令。

ra 输出所有寄存器。

f 输出浮点寄存器。

fd 输出双精度浮点寄存器。

b 输出当前子进程的所有断点、断点编号、相关计数、状态和命令。

ps 输出正在受 **adb** 跟踪的所有子进程的有关信息，即，进程 ID、类型（已创建或已连接）、计数和线程 ID。

pc 输出有关当前被调试的对象的信息。如果当前被调试的对象是子进程，则输出子进程信息（进程 ID、类型、计数），并输出有关该子进程的每个线程的信息（线程 ID、计数、信号）。如果当前被调试的对象是核心，则输出有关该核心中现有的每个线程的信息（线程号、utid、lwpid、PC 值和 PC 符号）。

pt 输出有关当前线程的信息（线程 ID、计数、信号）。如果当前被调试的对象是核心，则输出有关该核心当前线程的信息。（utid、lwpid 和寄存器信息）。

c 参数可以是 *address* 和 *count* 。输出 C 堆栈回溯跟踪。如果给定了 *address* ，则将其作为当前帧（而不是普通堆栈帧指针）的地址。如果给定了 *count* ，则仅输出前 *count* 个帧。

在向后兼容模式中，此命令具有非标准的不推荐行为。如果未给定参数，则此命令将使用 *address* 和 *count* 。

w [*width*] 将输出的页面宽度设置为 *width* （缺省值是 80。）

在向后兼容模式中，此命令具有非标准的不推荐行为。如果未给出 *width* ，则会将地址作为宽度。

s [*offset*] 将 **maxoffset** 设置为 *offset* 。

在向后兼容模式中，此命令具有非标准的不推荐行为。如果未给出 *offset* ，则会将地址作为偏移量。

o 将所有整数输入的缺省基数设置为八进制。

d [*radix*] 将缺省基数设置为 *radix* 。

在向后兼容模式中，此命令具有非标准的不推荐行为。如果未给定 *radix* ，则会将 *address* 作为基数。

x 将所有整数输入的缺省基数设置为十六进制。

q 退出 **adb** 。

v 输出所有变量的值。

m 输出地址映射。对于有效的 *corefile* ，它同时包含初始映射和缺省映射，并带有指出哪个映射为当前活动映射的指示。

z 输出信号列表，以及为当前子进程处理信号的方式。

zt 输出信号列表、相关标志以及为当前线程处理信号的方式。

k 输出所有 DLKM 模块或共享库。

n [*nodenumber*]
如果不使用参数，则输出 CCNUMA 计算机中的节点信息。如果使用 *nodenumber* 参数，则改为输出该节点的信息。

p *traditional_cmd*
此关键字命令会将传统命令作为参数，并对其进行解释。

a *var value* 将 *value* 赋给 **adb** 的变量 *var* 。

pa *Virtual_Offset*
以十六进制格式输出给定 *Virtual Offset* 的物理地址。空间 ID 取自 *adb* 的变量 *space* 。使用前面介绍的关键字命令 *a* ，可以设置 *adb* 的变量 *space* 。

下列命令只能在向后兼容模式中运行。

newline 输出进程 ID 和寄存器值。

M 在初始映射设置或有效内存文件与缺省映射对之间，切换内存文件的地址映射，用户可以使用文件操作修饰符 **>** 修改缺省映射对。如果内存文件无效，则只能使用缺省映射。

N [*nodenumber*]
输出 V 系列多节点计算机中的节点数，以及当前节点数。要切换到另一个节点，请输入 **$N nodenumber** 。

F 输出双精度浮点寄存器。

R 输出所有寄存器。

U 输出展开表。

格式字符串

“格式字符串”用于指定在由 **adb** 输出数据之前要进行的格式处理。 **adb** 支持两种类型的格式字符串： “传统样式”和 “**printf** 样式”。传统样式的格式字符串是一个 “格式说明符”序列。 **printf** 样式的格式字符串总是以逗号 (**,**) 开头，并用双引号 (**""**) 括起，它是格式说明符和其他字符的序列。每个格式说明符都应以一个 **%** 字符开头。非格式说明符的字符按原样输出。如果需要， **%** 应使用 **%** 进行转义。它支持 C 语言样式的 **** 字符转义序列。

处理格式字符串时， **adb** 会从左至右扫描格式字符串，并将每个遇到的 “转换说明符”应用到以 **dot** 和 **dotincr** 之和寻址到的对象。处理每个转换说明符之后， **dotincr** 将以 *count* 与该转换说明符的 *size* （隐式或显式）的乘积为增量增加。如果格式字符串用于输出 **dot** 的值（使用操作 **=** ），则 **dot** 和 **dotincr** 将保持不变。对于 **dotincr** 运算符，会相应地更新 **dotincr** 。

在向后兼容模式中，仅支持传统样式的格式字符串。

格式说明符

“格式说明符”可以是 “转换说明符”或 “点运算符”。

1. 转换说明符

每个转换说明符由以下部分组成：一个可选 *count* 或 *pspec* ，后跟一个可选 *size specifier character* ，再后跟一个 *conversion specifier character* 。

count 此选项仅适用于传统样式的格式字符串。 *count* 指定该转换说明符重复的次数。如果未指定，则假定 *count* 为 1。

pspec 此选项仅适用于 **printf** 样式的格式字符串。它是标志、字段宽度以及精度的序列，与 *printf*(3S) 库函数中相同。

size specifier character

此选项指定要应用此选项的对象的大小。可以用两种方式指定大小。一种是使用绝对大小说明符，另一种是使用相对大小说明符。绝对大小说明符如下所示。

b 对象的大小是 1 字节。

e 对象的大小是 2 字节。

g 对象的大小是 4 字节。

j 对象的大小是 8 字节。

k 对象的大小是 16 字节。

相对大小说明符如下所示

w 对象的大小是目标处理器的机器字大小。

h 对象的大小是目标处理器的机器字大小的一半。

l 对象的大小是目标处理器的机器字大小的二倍。

n 对象的大小是目标处理器的指针大小。对于宽文件和窄文件，此选项会不同。

m 对象的大小是目标处理器的指令的大小。仅在此选项为常数的处理器上才支持此选项。

Conversion Specifier Character

支持下列字符

a 以符号格式输出点的值。

c 以字符输出对象。

o 以无符号八进制数输出对象。

d 以有符号十进制数输出对象。

u 以无符号十进制数输出对象。

i 将对象反汇编为指令并输出。

f 根据对象大小，以浮点格式输出对象。

p 以符号格式输出对象。

s 假定对象为空终止字符串并输出。不能用于输出 **dot** 。

y 将对象转换为类型 **time_t** ，并以 *ctime*(3C) 格式输出对象。

这里， **printf** 样式的格式字符串仅支持 **c** 、 **o** 、 **d** 、 **u** 、 **x** 、 **f** 和 **s** 。如果未指定大小说明符字符，则有如下假定：对于转换字符 **c** 假定为 **b** ；对于转换字符 **d** 、 **u** 、 **x** 、 **o** 和 **f** 假定为 **w** ；对于 **i** 假定为 **m**；对于 **y** 假定为 **sizeof(time_t)** ；对于其他任何转换字符假定为 **w** 。

例如： **10=2bo, 'abc'=,"%s", main?4i**

2. 点运算符

一个点运算符由一个可选 *count* 、可选 *size specifier character* 以及一个 *dot operator character* 组成。

count *count* 指定该 *dot operator* 要重复的次数。如果未指定，则假定 *count* 为 1。对于 **printf** 样式的格式字符串， *count* 始终为 1。

Size Specifier Character
与转换说明符的大小说明符字符相同。

Dot operator character
可以是下列字符之一

v **dotincr** 以 *count* 倍大小为增量增加。

z **dotincr** 以 *count* 倍大小为减量减小。

例如： **=5bv, =5bv5bz**

向后兼容模式

在向后兼容模式中， “传统样式”可以是 *conversion specifier* 、 *dot operator* 、 *spacing specifier* 或 *literal string* 。

1. 转换说明符

一个转换说明符由一个可选 *count* 后跟一个 *conversion specifier character* 组成。

count 指定该 *conversion specifier* 要重复的次数。如果未指定，则假定 *count* 为 1。

Conversion Specifier Character

它们都有一个隐含大小。不识别显式大小。可使用下列格式字符： (在它们的旁边列出了其隐含大小)

o 2 以无符号八进制数输出对象。

O 4 以无符号八进制数输出对象。

q 2 以有符号八进制数输出对象。

Q 4 以有符号八进制数输出对象。

d 2 以有符号十进制数输出对象。

D 4 以有符号十进制数输出对象。

x 2 以无符号十六进制数输出对象。

X 4 以无符号十六进制数输出对象。

A 8 以无符号十六进制数输出对象。

u 2 以无符号十进制数输出对象。

U 4 以无符号十进制数输出对象。

f 4 以浮点数输出对象。

F 8 以双精度浮点数输出对象。

b 1 以十六进制数输出对象。

B 1 以八进制数输出对象。

c 1 以字符输出对象（忽略符号位）。

C 1 使用以下转义约定将对象作为字符输出。首先，忽略符号位，然后将字符值 000 至 040 输出为 @ 后跟在 0100 至 0140 范围内的相应字符。字符 @ 以 @@ 表示。

s *n* 假定对象为以 *n* 终止的字节序列，并作为指令进行输出。 *n* 的值是该指令占用的字节数。不能用于输出 **dot** 。

S *n* 假定对象为以空字节终止的字节序列。使用 **@** 转义约定将对象的这些字节输出为字符序列。 *n* 的值是对象中的字节数（包含空字节）。不能用于输出 **dot** 。

Y 4 以日期格式输出对象（请参阅 *ctime*(3C) ）。

i *n* 将对象反汇编为指令并输出。 *n* 的值是该指令占用的字节数。

a 0 以符号格式输出点的值。

p *n* 以符号格式输出对象。 *n* 的值与计算机相关。

例如： **main=ba, 'a'=c, main?10box**

2. 点运算符

一个点运算符由一个可选计数后跟一个点运算符字符组成。

count 与转换说明符的计数说明相同。

Dot operator character
点运算符字符可以是下列字符之一:

^ **dotincr** 以计数倍大小（与前面的转换说明符字符相对应）为减量减小。

+ **dotincr** 以计数为增量增大。

- **dotincr** 以计数为减量减小。

例如： **10=-, 10=2-, 10=5o4^**

3. 间隔说明符

一个间隔说明符由一个可选 *count* 或可选 *tabstop* 后跟一个 “间隔说明符字符”组成。

count 与转换说明符的计数说明相同。

tabstop
与转换说明符的计数说明相同。但是，它只能与 *t* 间隔说明符一起使用。如果没有给定任何值，则假定它为 1。

spacing specifier character
间隔说明符字符可以是下列字符之一:

t 移动至与制表位对应的下一个制表位。例如， **8t** 会移动至下一个 8 空格制表位。

r 输出一个空格。

n 输出一个换行符。

例如： **10=2o2t2o, 10=2o2r2o, 10=2o2n2o**

4. 文字字符串

文字字符串是用双引号 ("") 括起来的任何数量的字符。

例如： **10="in octal "ot"in hex "x**

地址映射

在对象文件和应用程序核心文件之类的文件中，虚拟内存地址不同于文件地址偏移量。因此， **adb** 维护着这些文件的一个 "地址映射" 数组，以便将给定的虚拟内存地址映射到文件地址偏移量。每个地址映射是一个三元组：起始虚拟地址 (*b*) 、结束虚拟地址 (*e*) 和起始文件地址偏移量 (*f*)。该三元组指定，从 *b* 至 *e* - 1 的所有 *addresses* 占用 *file* 中以 *f* 起始的连续区域。假设给定一个虚拟地址 *a* ，使 $b \leq a < e$ ，则 a 的文件地址偏移量可以这样计算： $f + a - b$ 。

状态变量

有几个变量定义了 **adb** 的在任何时刻的即时状态。它们是：

dot	当前地址。初始值为 0。
dotincr	当前地址增量。初始值为 0。
prompt	adb 使用的提示字符串。初始值是 "**adb>** "。
radix	当前输入基数。初始值与目标处理器的汇编语言中一致。
maxwidth	最大显示宽度。初始值是 80。
maxoffset	如果来自已知符号的地址在此限制范围之内，则 **adb** 将地址输出为 *symbol_name+offset* ，否则输出该 *address* 。初始值是 **0xffffffff** 。
macropath	用于搜索 **adb** 宏的目录列表。初始值是 **.:/usr/lib/adb** 。
pager	adb 使用的 pager 命令。初始值是 **more -c** 。
backcompat	如果 **adb** 处于向后兼容模式下，则设置为 1。初始值取决于主机处理器。

注释

adb64 是 **adb** 的符号链接。在向后兼容模式中，使用一些旧脚本维护此符号链接，这些旧脚本可能要使用 **adb64**。

外部语言环境影响

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

adb 对无法访问的文件、语法错误、命令异常终止等做出了注释。除非最后一个命令失败或返回非零状态，否则退出状态为 **0** 。

作者

adb 由 HP 开发。

文件

a.out

core

/dev/mem

/dev/kmem

另请参阅

ttrace(2)、 crt0(3)、 ctime(3C)、 end(3C)、 a.out(4)、 core(4)、 signal(5)。

《ADB Tutorial》

名称

adjust - 简单的文本格式化程序

概要

adjust [**-b**] [**-c**|**-j**|**-r**] [**-m** *column*] [**-t** *tabsize*] [*files* ...]

说明

adjust 命令是一种简单的文本格式化程序，可用于填充、居中、左右对齐或仅右对齐文本段落，并用于交互使用。它读取一连串的输入文件（或者，如果没有指定输入文件则使用标准输入），然后在标准输出上生成对其输入格式的改良版本，每个段落都单独格式化。如果将 **-** 指定为输入文件名，则 **adjust** 在该点读取标准输入（使用 **--** 作为参数来将 **-** 各个选项分隔开）。

adjust 将输入行中的文本读取为由空格、制表符或换行符分隔的一系列字。将文本行进行分组而划分成段落，段落之间由空行进行分隔。缺省情况下，文本直接复制到输出，只遵守简单填充（如下所示），右间距为 72，并且前导空格转换为制表符（如有可能）。

选项

adjust 命令可识别下列命令行选项：

-b 不会在输出上将前导空格转换为制表符；（输出中不包含制表符，即使输入中有制表符，亦如此）。

-c 将每行文本居中。行是经过预处理和后处理的，但不执行填充。

-j 对齐文本。在填充之后，根据需要在每行插入空格实现右对齐（而每一段的最后一行除外），同时保持已对齐的左边距。

-r 填充文本后，调整每一行的缩进，以实现对齐右边距（左边距未对齐）。

-m*column* 将右侧填充边距设置为指定的列数，而不是 72。填充文本并适当右对齐，这样输出行不会超过该列（如有可能）。如果指定 **-m0**，则每一段第一行当前的右边距用于该行以及该段落中的所有后续行。

缺省情况下，文本在第 40 列上居中。使用 **-c**， **-m** 选项将设置居中“窗口”的中间列，但 **-m0** 如前所述自动设置右边距（然后确定“窗口”的中心）。

-t*tabsize* 将制表符大小设置为非缺省值（八列）。

在一个命令行中只允许 **-c**、 **-j** 和 **-r** 选项中的一个选项。

详细信息

在对输入文本的行进行其他操作之前， **adjust** 首先处理退格符，以常用方式擦除前导字符。接下来，它忽略除制表符以外的所有不可打印的字符。然后，将所有制表符扩展为空格。

对于简单的文本填充，每一段第一行的第一个字与输入行具有相同的缩进量。然后，每个字都放入到输出，并在后面紧跟一个空格。以 *terminal_character*[*quote*][*closing_character*] 结尾的“单词”后面紧跟两个空格，其中，*terminal_character* 是 **.**、 **:**、 **?** 或 **!** 中的任一字符； *quote* 是一个单结束引号 (**'**) 字符或双引号字符 (**"**)，而

closing_character 是 **)** 、 **]** 或 **}** 中的任一字符。以下是一些示例:

end. of? sentence.' sorts!" of.) words?"]

（**adjust** 不会在 *terminal_character* 后面的一对单结束引号 (**"**) 之后放置两个空格）。

adjust 每当向当前行添加一个字（不是第一个字）将超出右边距时，都将开始一个新输出行。

填充文本时 **adjust** 可以识别段落（例如本段落）的首行缩进格式。每一段的第二行及后续行都与输入段落的第二行具有相同的缩进量（如果有第二行），否则与第一行的缩进量相同。

***** 填充文本时 **adjust** 对带标记的段落（例如本段落）有初步识别。如果段落第二行的缩进量大于第一行，而且第一行有一个字在与第二行相同缩进处开始，则保留标记字的输入列位置（在匹配第二行缩进的字之前）。

标记字的位置可忽略而不会更改列位置，即使这些标记字超过右边距，亦如此。填充行的其余部分，或者从第一个非标记字的位置右对齐行的其余部分。

当指定 **-j** 时， **adjust** 使用智能算法在最需要的输出行中插入空格，直到行扩展到右边距。首先，检查所有的单空格字分隔符。向每个分隔符中添加一个空格，以该分隔符以及前面和后面的分隔符之间具有最多字母的空格开始，直到修改的行达到右边距。如果将所有的单空格分隔符增加到两个空格，且必须插入更多空格，则使用两空格分隔符重复算法，依此类推。

输出行缩进保持小于右边距。如果一个字大于行大小（右边距减去缩进），则该字会单独显示在一行上，正确缩进，而且超过右边距。但是，如果使用 **-r** ，这些字仍然会右对齐（如果可能）。

如果当前的语言环境定义了类名称 **ekinsoku** 和 **bkinsoku** （请参阅 *iswctype*(3C) ），则 **adjust** 根据 **ekinsoku/bkinsoku** 字符分类和边距设置（请参阅 **-r** 、 **-j** 和 **-m** 选项）来格式化文本。

外部语言环境影响

环境变量

LANG 为没有设置或设置为 null（空）的国际化变量提供了缺省值。如果未设置 **LANG** 或为空，则使用缺省值 “C” （请参阅 *lang*(5) ）。如果任一国际化变量中包含无效设置，则 **adjust** 就会认为所有国际化变量都设置为 “C”。请参阅 *environ*(5) 。

LC_ALL 如果设为非空字符串值，则会覆盖所有其他国际化变量的值。

LC_CTYPE 确定文本解释为单字节和（或）多字节字符，确定字符分类为可打印字符，以及确定正则表达式中由字符类表达式匹配的字符。

LC_MESSAGES 确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息以及写入到标准输出的信息消息的格式和内容。

NLSPATH 决定了处理 **LC_MESSAGES** 的消息目录所在的位置。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

如果无法打开任何输入文件（它将跳过该文件），则 **adjust** 报告到标准错误，并在稍后返回一个非零值。如果参数 **-m** 或 **-t** 超出范围，或者没有正确调用程序，则会执行同样的操作（但会立即退出）。

大于 **BUFSIZ** 的输入行会自动拆分（在制表符扩展之前）或截断（在制表符扩展之后）。太宽从而无法居中的行从列 1 开始（没有前导空格）。

举例

对于在 *vi*(1) 中进行文本过滤，该命令非常有用。例如，

!}adjust

重新格式化当前段落的其余部分（从当前行以下），均匀排列各行。

vi 命令：

:map ^X {!}adjust -j^V^M

（其中 ^ 表示控制字符）设置有用的“指针宏”。键入 **^X** (Ctrl-X) 重新格式化整个当前段落。

adjust -m1 可以方便地将文本分成单独的字，除了已标记的段落标记外，没有空白。

警告

该程序旨在简单而且快速。它不识别反斜杠来转义空格或其他字符。它不识别那种标记单独位于一行的带标记段落。它知道行以换行符或 Null（空）结尾，以及如何处理制表符和退格符，但却不对诸如换页符之类的其他字符进行任何特殊操作（这些字符简单地被忽略）。对于复杂操作，标准文本处理器可能更适合。

该程序还可以用一组独立的程序来实现：填充、居中和对齐（使用 **-r** 选项时）。但在实际使用中，这样做效率会更低，特别是考虑到该程序对标记段落和段落的最后几行能够进行专门的识别和处理时。

作者

adjust 由 HP 开发。

另请参阅

nroff(1)。

名称

admin - 创建和管理 SCCS 文件

概要

admin -i[*name*] [**-n**] [**-b**] [**-a** *login*] ... [**-d** *flag[flag-val*]] ... [**-f** *flag[flag-val*]] ...
[**-m** *mrlist*] ... [**-r** *rel*] [**-t**[*name*]] [**-y**[*comment*]] *file* ...

admin -n [**-a** *login*] ... [**-d** *flag[flag-val*]] ... [**-f** *flag[flag-val*]] ... [**-m** *mrlist*] ...
[**-t**[*name*]] [**-y**[*comment*]] *file* ...

admin [**-a** *login*] ... [**-e** *login*] ... [**-d** *flag[flag-val*]] ... [**-m** *mrlist*] ...
[**-r** *rel*] [**-t**[*name*]] *file* ...

admin -h *file* ...

admin -z *file* ...

说明

admin 命令用于创建新的 SCCS 文件以及更改现有 SCCS 文件的参数。 **admin** 的参数可以按任何顺序出现（除非将 **--** 指定为参数，在这种情况下 **--** 后面的所有参数都将被视为文件），由以 **-** 开头的选项参数和指定名称的 *file*（请注意，SCCS 文件名必须以字符 **s.** 开头）组成。如果指定的 *file* 不存在，则将创建它，并根据指定的选项参数初始化其参数。为选项参数没有初始化的参数分配缺省值。如果指定的 *file* 确实存在，则将更改与指定的选项参数对应的参数，而其他参数保持不变。

如果指定了 *directory* 而不是 *file* ，则 **admin** 作用于 *directory* 中的每个 *file* ，但是其中非 SCCS 文件（路径名中不以 **s.** 开头的最后部分）和不可读文件将被忽略（不出现任何用户提示）。如果指定的名称是 **-** ，则会读取标准输入，并将标准输入的每一行假定为要处理的 SCCS 文件的名称。此外，仍将忽略（不出现任何用户提示）非 SCCS 文件和不可读文件。

admin 选项参数独立地应用于所有指定 *file* ，而不管是一个文件还是多个文件。在下面的讨论中，虽然好像仅对一个指定文件的每个选项进行了说明，但这些选项对单个或多个文件的影响是相同的。

选项

admin 命令支持以下选项和命令行参数：

-n 此选项指明要创建的新 SCCS 文件。

-i[*name*] 指文件的 *name* ，新的 SCCS 文件的内容从此文件中获得（如果由 *name* 指定的文件为二进制文件，则必须指定 -b 选项）内容构成文件的第一个 delta 版（有关 delta 版的编号方案，请参阅 **-r** 选项）。如果使用 **-i** 选项但省略文件名，则读取标准输入中的文本直至遇到文件结束标志。如果省略这个选项，创建的 SCCS 文件将具有空的初始 delta 版。使用带有 **-i** 选项的 **admin** 命令只能创建一个 SCCS 文件。使用单个 **admin** 创建两个或更多 SCCS 文件要求将它们创建为空文件（无 **-i** 选项）。请注意， **-i** 选项隐含 **-n** 选项。

-b 对 **-i** 选项所指定的名称的内容进行编码。如果文件名称是二进制文件，则必须使用这个选项关键字；否则， SCCS 命令将无法正确处理二进制文件。

-r *rel* 指初始 delta 版插在其后的的版本号 (*rel*) 。仅当也使用 **-i** 选项时才可以使用此选项。如果未使用 **-r** 选项，则初始 delta 版将插入到版本号 1 之后。初始 delta 版的级别始终为 1 （缺省情况下，初始 delta 版为 1.1）。

-t[*name*] 指文件的 *name* ，即 SCCS 文件的描述性文本从该文件中提取。如果使用 **-t** 选项，且 **admin** 正在创建新的 SCCS 文件（也使用 **-n** 和（或） **-i** 选项），则还必须提供描述性文本的文件名。对于现有 SCCS 文件：

- 如果 **-t** 选项不带文件名，则会导致删除当前包含在 SCCS 文件中的描述性文本（如果有的话）。
- 如果 **-t** 选项带有文件名，则会导致用指定文件中的文本（如果有的话）替换当前包含在 SCCS 文件中的描述性文本（如果有的话）。

-f *flag* 此选项指定要放置在 SCCS 文件中的一个 *flag* ，还可能指定该 *flag* 的值。在一个 **admin** 命令行中，可以使用多个 **-f** 选项。允许的 *flag* 及其值如下所示：

b 允许在 **get** 命令上使用 **-b** 选项（请参阅 *get*(1) ）以创建分支 delta 版。

c*ceil* 最高版本（即“上限”），小于或等于 9999 的编号，可以由 **get** 命令检索以进行编辑。未指定的 **c** 标志的缺省值为 9999。

f*floor* 最低版本（即“下限”），大于 0 且小于 9999 的编号，可以由 **get** 命令检索进行编辑。未指定的 **f** 标志的缺省值为 1。

d*SID* 由 **get** 命令使用的缺省 delta 版编号 *SID* （请参阅 *get*(1) ）。

i*str* 产生消息：

No id keywords (cm7)

由 **get** 或 **delta** 发出，被视为致命错误（请参阅 *delta*(1) ）。如果缺少此标志，则消息仅作为警告类消息。如果在 SCCS 文件中检索或存储的文本中找不到 SCCS 标识关键字（请参阅 *get*(1) ），则将发出消息。如果提供了值，则关键字必须与提供的字符串完全匹配。但是，字符串必须包含关键字，但不得包含嵌入的换行符。

j 允许并发使用多个 **get** 命令，以便对 SCCS 文件的同一 *SID* 进行编辑。这样，就可以对 SCCS 文件的同一版本进行多个并发更新。

只有一个用户可以执行并发编辑。多用户访问通常是通过使用公共登录名或设置用户 ID 程序实现的（请参阅 *chmod*(1) 和 *exec*(2) ）。

l*list* 不再改变其 delta 版的版本 *list* （对其中一个被锁定的版本执行 **get -e** 将失败）。 *list* 具有以下语法:

list **::=** *range* | *list* **,** *range*
range **::=** *RELEASE NUMBER* | **a**

list 中的字符 **a** 等效于为指定的 SCCS 文件指定 “所有版本” 。省略任何列表等效于 **a** 。

n 导致在 新版本中产生 delta 版时 **delta** 将在要跳过的每个版本（如果有的话）中创建空 delta 版（例如，在 delta 版 2.7 之后产生 delta 版 5.1 时，将跳过版本 3 和版本 4）。这些空 delta 版充当 “定位点”，以便稍后可以从其创建分支 delta 版。缺少此标志会导致 SCCS 文件中不存在所跳过的版本，防止将来从其创建分支 delta 版。

q*text* 用户定义的文本，用于替换通过 **get** 命令检索的 SCCS 文件文本中出现的所有 **%Q%** 关键字。

m*mod* SCCS 文件的 *module* 名称，用于替换通过 **get** 命令检索的 SCCS 文件文本中出现的所有 **%M%** 关键字。如果未指定 **m** 标志，则分配的值是删除了前导 **s.** 的 SCCS 文件名称。

t*type* SCCS 文件中模块的 *type* ，用于替换通过 **get** 命令检索的 SCCS 文件文本中出现的所有 **%Y%** 关键字。

v[*pgm*] 导致 **delta** 提示输入修改请求 (*MR*) 编号作为创建 delta 版的原因。选择的值可指定 (*MR*) 编号有效性检查程序的名称（请参阅 *delta*(1) ）（如果在创建 SCCS 文件时设置了此标志，则即使其值为空值，也必须使用 **m** 选项）。

x 可使 **get** 命令创建带有执行权限的文件。

-d *flag* 导致从 SCCS 文件中去除（删除）指定的 *flag* 。仅当处理现有 SCCS 文件时，才能指定 **-d** 选项。在一个 **admin** 命令行上，可以使用多个 **-d** 选项。有关允许的 *flag* 名称，请参阅 **-f** 选项。

l*list* 要解锁的版本的 *list* 。有关 **l** 标志和 *list* 语法的说明，请参阅 **-f** 选项。

-a *login* 要添加到用户列表的 *login* 名或数字型 HP-UX 组 ID，允许这些用户对 SCCS 文件进行 delta（变更）。指定组 ID 相当于指定该组 ID 的所有公共 *login* 名。在一个 **admin** 命令行上，可以使用多个 **a** 选项。可以同时在列表中提供所需数目的 *login* 或数字型组 ID。如果用户列表为空，则任何人都可以添加 delta 版。前面有 **!** 的 *login* 或组 ID 拒绝产生 delta 版的权限。

-e *login* 要从用户列表中清除的 *login* 名或数字型组 ID，允许这些用户对 SCCS 文件进行 delta（更改）。指定组 ID 等效于指定该组 ID 的所有公共 *login* 名。在一个 **admin** 命令行

上，可以使用多个 **e** 选项。

-y[*comment*] *comment* 文本作为初始 delta 版的注释插入到 SCCS 文件中，插入方式与 *delta*(1) 的相同。省略 **-y** 选项，将插入以下形式的缺省注释行：

date and time created *YY* **/** *MM* **/** *DD* **/** *HH* **/** *MM* **/** *SS* **by** *login*

仅当指定 **-i** 和（或） **-n** 选项（即创建新的 SCCS 文件）时， **-y** 选项才有效。

-m *mrlist* 作为创建初始 delta 版的原因将修改请求 (*MR*) 编号的列表插入到 SCCS 文件中，插入方式与 *delta*(1) 的相同。必须设置 **v** 标志，并且如果 **v** 标志具有值（即 (*MR*) 编号验证程序的名称），则检验 (*MR*) 编号。如果未设置 **v** 标志或 (*MR*) 验证失败，则将出现诊断消息。

-h 可使 **admin** 检查 SCCS 文件的结构（请参阅 *sccsfile*(4) ），并将新计算的校验和（ SCCS 文件中除首行中字符之外的所有字符的总和）与存储在 SCCS 文件的首行中存储的校验和进行比较。将生成相应的错误诊断信息。

此选项禁止对文件进行写操作，这样就消除了任何其他选项所产生的影响，因此仅在处理现有文件时有意义。

-z 重新计算 SCCS 文件校验和，并将其存储在 SCCS 文件的首行中（请参阅上面的 **-h** ）。

请注意，对确实已损坏的文件使用此选项可以防止将来检测到损坏。

访问控制列表 (ACL)

不要将可选的 ACL 条目添加到 SCCS 文件。 SCCS 会删除它们，可能导致意外的和不需要的访问模式。

外部语言环境影响

环境变量

LC_CTYPE 用于确定将文本解释为单字节字符和（或）多字节字符。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或 **LC_MESSAGES** ，或者将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值 **C** （请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **admin** 就会认为所有国际化变量都设置为 **C** 。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

使用 *sccshelp*(1) 进行说明。

警告

SCCS 文件中的行数不能超过 99,999 行。在非 XPG4 环境中每行可以包含 **BUFSIZ**-3 个字符，在 XPG4 环境中每行可以包含 **LINE_MAX**-3 个字符（包括换行符）。

文件

所有 SCCS 文件名的最后部分必须采用 **s.***filename* 形式。对新建的 SCCS 文件提供 444 权限模式（请参阅 *chmod*(1)）。需要有相关目录的写入权限才能创建文件。由 **admin** 执行的所有写入操作是写入名为 **x.***filename* 的临时 x 文件（请参阅 *get*(1)）。如果 **admin** 命令创建的是一个新的 SCCS 文件，则该临时文件被赋予权限模式 444；如果 **admin** 命令执行的是一个现有的 SCCS 文件，则该临时文件被赋予与该 SCCS 文件相同的权限模式。成功执行 **admin** 后，将删除 SCCS 文件（如果它存在），并将 x 文件重命名为 SCCS 文件的名称。这样就确保仅当没有出现错误时，才对 SCCS 文件进行更改。

建议对包含 SCCS 文件的目录赋予权限模式 755， SCCS 文件本身采用权限模式 444。任何给定目录的模式仅允许所有者修改该目录中包含的 SCCS 文件。 SCCS 文件的模式可防止除 SCCS 命令之外的所有命令进行的任何修改。

如果由于任何原因需要修补 SCCS 文件，则所有者可以将模式更改为 644，这样就允许使用 **vi** 或任何其他合适的编辑器。 必须当心！对已编辑的文件应该 总是先执行 **admin -h** 命令来检查是否有损坏，然后执行 **admin -z** 命令生成正确的校验和。另外建议使用 **admin -h** 命令来确保 SCCS 文件是有效的。

admin 还使用名为 **z.***filename* 的暂态锁文件，该文件用于防止不同用户同时对 SCCS 文件进行更新。有关进一步的信息，请参阅 *get*(1)。

另请参阅

delta(1)、 ed(1)、 get(1)、 sccshelp(1)、 prs(1)、 what(1)、 sccsfile(4)、 acl(5)。

符合的标准

admin：SVID2、SVID3、XPG2、XPG3、XPG4

名称

answer - 电话消息转录系统

概要

answer [-pu]

说明

使用 **answer** 交互式程序可将电话（和其他）消息转录到电子邮件中。

该程序使用您个人的 **elm** 别名数据库和系统 **elm** 别名数据库，允许您使用别名为消息提供地址。

选项

answer 支持下列选项：

-p 提示输入电话单类型消息字段。

-u 允许使用非别名的地址。

操作

answer 以 **Message to:** 提示开始。输入含一个单词的别名或含两个单词的用户名（“单词”之间由空格分隔）。用户名被转换为 *f_lastword* 形式的别名，其中 *f* 是第一个单词的首字符， *lastword* 是第二个单词，并且所有字母都转换为小写。例如， **Dave Smith** 将转换为别名 **d_smith** 。

如果不使用 **-u** 选项，则指定或转换的别名必须存在于数据库中。在使用 **-u** 选项的情况下，如果处理的“别名”不在别名数据库中，则将使用该别名本身作为地址。

将显示完全扩展地址。

在使用 **-p** 选项的情况下，将要求您输入典型的消息单数据：

Caller:
of:
Phone:

TELEPHONED -
CALLED TO SEE YOU -
WANTS TO SEE YOU -
RETURNED YOUR CALL -
PLEASE CALL -
WILL CALL AGAIN -
*******URGENT****** -**

输入相应的数据。在相关短线提示符后，可以仅输入一个 **X** 或不输入任何内容，还可以键入较长的注释。所输入的任何内容将成为消息的一部分。将忽略消息中未添加任何文本的行。

最后，提示您输入消息。如果有的话，输入消息，并以空行结束。将发送消息并重复出现 **Message to:** 提示。

要结束程序，请在 **Message to:** 提示后输入 **bye** 、 **done** 、 **exit** 或 **quit** 之一。

举例

用户输入是正常类型。

不使用任何选项

该示例显示有一个有效别名、一个无效用户名和一个有效用户名。在别名无效的情况下，转换的别名将显示在方括号中。

--

Message to: oswald
address 'oswaldr@mycompany.com (Oswald Rarebit)'

Enter message for oswald ending with a blank line.

> And here is the message.
>

--

Message to: albert einstein
Sorry, could not find 'albert einstein' [a_einstein] in list!

Message to: john jones
address 'mrjones@companybee.com (John P. Jones)'

Enter message for john jones ending with a blank line.

> A nice message
>

--

Message to: quit

使用了 **-u** 选项

如果输入 **answer -u** ，则将以不同的方式处理前面的错误。

--

Message to: albert einstein

address ’a_einstein’

Enter message for albert einstein ending with a blank line.

> About your theory ...
>

使用了 **-p** 选项

如果输入 **answer -p** ，则会添加电话单提示符。将从消息中删除未添加任何文本的三行。

--

Message to: George Dancer
address ’g_dancer@cup.hp.com (George B. Dancer)’

Caller: Harold Maudlin
of: Terpsichore Inc.
Phone: 123 456 7890

TELEPHONED - at 4:30pm
CALLED TO SEE YOU -
WANTS TO SEE YOU - X
RETURNED YOUR CALL -
PLEASE CALL - X
WILL CALL AGAIN -
*******URGENT****** -** very very!

Enter message for George Dancer ending with a blank line.

> He said that you would ...
>

文件

$HOME/.elm/aliases	用户别名数据库数据表
$HOME/.elm/aliases.dir	用户别名数据库目录表
$HOME/.elm/aliases.pag	用户别名数据库散列表
$HOME/.elm/aliases.text	用户别名源文本
/var/mail/.elm/aliases	系统别名数据库数据表
/var/mail/.elm/aliases.dir	系统别名数据库目录表
/var/mail/.elm/aliases.pag	系统别名数据库散列表

/var/mail/.elm/aliases.text	系统别名源文本
/tmp/snd.*pid*	传出邮件消息编辑缓冲区

作者

answer 由 HP 开发。

另请参阅

elm(1)、 newalias(1)。

名称

ar - 创建并维护可移植归档及库

概要

ar [-]*key* [-][*modifier* ...] [*posname*] *afile* [*name* ...]

说明

ar 命令可维护合并为一个单独的归档文件的文件组。它的主要用途是创建并更新库文件，与链接编辑器使用它时相同（请参阅 *ld*(1)）。但它也可以用于实现任何类似的目的。 **ar** 使用的 magic 字符串和文件头是由可输出的 ASCII 字符构成的。如果一个归档是由多个可输出文件组成的，则整个归档也是可输出的。

单个文件可以插入，但不转换为归档文件。当 **ar** 创建归档时，它会采用一种可跨所有计算机移植的格式创建文件头。有关可移植归档格式和结构的详细说明，请参阅 *ar*(4)。链接编辑器使用归档符号表（如 *ar*(4) 中所述）在对象文件库中重复且有效地进行搜索。仅当归档至少包含一个对象文件时， **ar** 才创建并维护归档符号表。归档符号表位于一个以特殊方式命名的文件中，该文件始终是归档中的第一个文件。该文件永远不会被用户所提及或访问。每当使用 **ar** 创建或更新归档的内容时，将会重建符号表（除非使用了 **z** 修饰符）。下面介绍的 **s** 修饰符可强制执行符号表的重建。

drqtpmx 集合中的 *key* 操作字符是必需的，该字符可选择以连字符 (**-**) 开头。所需的 *key* 操作字符可以使用 **abcfFilsuvzACT** 集合中的一个或多个 *modifier* 字符进行指定。将 *posname* 与 **r** 和 **m** 关键字操作以及 **a**、 **b**、 **i** 修饰符一起使用，可以指定归档中的位置。 *afile* 是归档文件。归档文件的组成文件是通过 *name* 参数指定的。

下表介绍了 *key* 操作字符：

d 从归档文件中删除命名文件。

r 替换命名文件，或向归档中添加新文件：

- 如果将 **u** 修饰符与操作字符 **r** 一起使用，则只有修改日期晚于相应成员文件的文件会被替换。
- 如果使用了 **abi** 集合中的可选定位字符，则必须提供 *posname* 参数，并指定新文件将放在 (**a**) 之后或者（b 或 **i**） *posname* 之前。如果没有定位字符，则新文件将放在结尾处。
- 如果不存在 *afile*，则 **ar** 将创建它。
- 如果未指定 *name*，并且：
  - 指定的归档文件不存在，则 **ar** 将创建一个仅包含归档头的空归档文件（请参阅 *ar*(4)）。
  - 归档中包含一个或多个其名称与当前目录中文件名相匹配的文件，则每个匹配的归档文件将被替换为相应的本地文件，而不考虑哪个文件比较新，除非还指定了 **u** 修饰符。

q 将命名文件快速附加到归档文件的结尾。定位字符无效。该操作不检查添加的成员是否已在归档中。如果不存在 *afile*，则 **ar** 将创建它。

t 将归档文件的目录表输出到标准输出中。如果未给定名称，则显示归档中的所有文件。如果给定了名称，则仅显示这些文件的信息。

p 将归档中的命名文件输出到标准输出中。如果未指定任何名称，则所有文件的内容按这些文件在归档中出现的顺序输出。

m 移动命名文件。缺省情况下，文件移动到归档的结尾处。如果提供了定位字符，则必须同时提供 *posname* 参数，与 **r** 操作中相同， *posname* 参数指定移动文件的目标位置。请注意，在与定位字符一起使用时，文件会按它们当前在归档中出现的同一顺序进行移动，而 不是按命令行中的指定顺序。请参阅“举例”。

x 解压缩命名文件。如果未给定名称，则归档中的所有文件将被解压缩。在任何情况下， **x** 都不能更改归档文件中的条目。

下表介绍了可选的 *modifier* 字符：

a 将文件放在由 *posname* 指定的现有定位文件之后。

b 将新文件放在由 *posname* 指定的现有定位文件之前。

c 禁止显示通常在创建 *afile* 时生成的消息。对于 **r** 和 **q** 操作而言，如果不存在 *afile* ，则 **ar** 通常会创建它。

f 在对归档执行操作之前，将命名文件名截断为 14 字节。提供该修饰符的目的是与先前的版本兼容，在先前的版本中支持文件名最多为 14 字节。长文件名会被截断。当与 **r** 操作一起使用时，与截断后的文件名匹配的第一个现有文件将被替换。也可以将 **f** 修饰符与其他操作一起使用，以允许指定完整的文件名，而不是截断的文件名。另请参阅 **F** 修饰符的说明。

i 将新文件放在由 *posname* 指定的现有定位文件之前。与 **b** 修饰符相同。

l 将临时文件放在本地当前工作目录中，而不是由环境变量 **TMPDIR** 指定的目录或者缺省目录 **/var/tmp** 。只有 **d** 、 **m** 、 **q** 和 **r** 操作以及 **s** 和 **F** 修饰符会使用临时文件。

s 重新生成归档符号表，即使 **ar** 未调用修改归档内容的操作也是如此。在归档上使用 **strip** 命令（请参阅 *strip*(1) ）后或使用 **z** 修饰符修改归档后，若要恢复归档符号表，则该修饰符非常有用。

u 更新归档（仅用于 r 操作）不要向归档中复制本地文件，除非本地文件比归档中相应的现有文件要新。

v 在标准输出中逐文件给出有关创建或修改归档文件的详细说明。当与 **t** 、 **v** 一起使用时，会给出有关文件的所有信息的长列表。当与 **d** 、 **m** 、 **p** 、 **q** 或 **x** 操作一起使用时，verbose 修饰符会让 **ar** 输出与该操作相关联的每个 *key* 操作字符和文件名。对于 **r** 操作，如果添加一个新文件，则 **ar** 显示 **a** ；如果替换现有文件，则显示 **r** 。对于 **p** 操作， **ar** 会在输出文件内容之前在标准输出中输出文件名。

z 禁止在修改归档时重建符号表。当逐块创建大型归档时，该修饰符仅对避免花费较长的构建时间有用。如果现有归档包含一个符号表，则 **z** 修饰符会使其无效。如果给定一个长度超过 15 字节的文件名，则整个归档将被重写。要重建符号表，请使用 **ranlib** 命令（请参阅 *ranlib*(1) ），或重新调用带有 **s** 修饰符的 **ar** 。

A 禁止显示有关可选访问控制列表条目的警告消息。 **ar** 不会对文件访问控制列表中的可选访问控制列表条目进行归档（请参阅 *acl*(5) ）。通常情况下，对于每个包含可选访问控制列表条目的文件，输出一条警告消息。

C 防止解压缩的文件被同名文件替换。 **C** 修饰符只能与 **x** 操作一起使用。

F 截断整个归档。 **F** 修饰符使整个归档被重写，因此归档中的所有文件名都截断为 14 字节，即使当 **ar** 不对归档内容进行修改时也是如此。长名称的表将被删除（请参阅 *ar*(4) ）。提供该修饰符的目的是与先前的版本兼容，在先前的版本中支持文件名最多为 14 字节。另请参阅 **f** 修饰符的说明。

T 将其归档名的长度大于文件系统支持长度的文件名截断。缺省情况下，文件名长度大于文件系统支持长度的文件不会被解压缩，并会产生错误。 **T** 修饰符只能与 **x** 操作一起使用。

只有下列组合是有意义的；其他修饰符与操作的组合对操作没有任何影响：

d : **v** 、 **f** 、 **F** 、 **l**
m : **v** 、 **f** 、 **F** 、 **l** 与 **a** | **b** | **i**
r : **v** 、 **f** 、 **F** 、 **l** 、 **c** 、 **A** 、 **u** 与 **a** | **b** | **i**
q : **v** 、 **f** 、 **F** 、 **l** 、 **c** 、 **A** 、 **z** 、 **s**
t : **v** 、 **f** 、 **F** 、 **s**
p : **v** 、 **f** 、 **F** 、 **s**
x : **v** 、 **f** 、 **F** 、 **s** 、 **C** 、 **T**

外部语言环境影响

环境变量

下列国际化变量会影响 **ar** 的执行：

LANG 确定没有 **LC_ALL** 和其他 **LC_*** 环境变量时本地语言、本地惯例和编码字符集的语言环境类别。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值 **C** （请参阅 *lang*(5) ），而不是 **LANG** 。

LC_ALL
确定所有语言环境类别的值，它优先于 **LANG** 和其他 **LC_*** 环境变量。

LC_CTYPE
确定字符处理函数的语言环境类别。

LC_MESSAGES
确定应该用来影响写入标准错误的诊断消息的格式和内容的语言环境。

LC_NUMERIC
确定数字格式的语言环境类别。

LC_TIME
确定日期和时间格式化时的格式和内容。

NLSPATH

确定用于处理 **LC_MESSAGES** 的消息目录的位置。

如果任一国际化变量包含无效设置，则 **ar** 就会认为所有国际化变量都设置为 **C** 。请参阅 *environ*(5) 。

此外，以下环境变量影响 **ar** ：

TMPDIR

指定用于保存临时文件的目录（请参阅 *tmpnam*(3S) ）。 **l** 修饰符会覆盖 **TMPDIR** 变量， **TMPDIR** 会覆盖缺省目录 **/var/tmp** 。

诊断信息

phase error on *file name*

在 **ar** 将命名文件复制到归档中时，命名文件被另一个进程修改。发生这种情况时， **ar** 会退出，原始归档保持原样。

ar write error: file system error message

ar 不能写入临时文件或最终输出文件。如果 **ar** 试图写入最终输出文件，则原始归档会丢失。

ar 报告 **cannot create** *file*.**a**，其中 *file*.**a** 是 **ar**格式的归档文件，即使 *file*.**a** 已经存在也是如此。当 *file*.**a** 处于写保护状态或不可访问时，就会触发该消息。

举例

以归档格式创建一个新文件（如果不存在这样一个文件），按指定顺序输入其成员文件：

ar r newlib.a f3 f2 f1 f4

替换文件 **f2** 和 **f3** ，使新副本位于文件 **f1** 之后， **f3** 位于 **f2** 之后：

ar ma f1 newlib.a f2 f3
ar ma f2 newlib.a f3
ar r newlib.a f2 f3

随后对归档进行排序：

newlib.a: f1 f2' f3' f4

其中，单引号表示更新的文件。第一条命令的意思是：“将 **f2** 和 **f3** 移到 *newlib*.a 中的 **f1** 之后”，因而创建排序：

f1 f3 f2 f4

请注意， **f2** 和 **f3** 的相对顺序未改变。第二条命令的意思是：“将 **f3** 移到 *newlib.a* 中的 **f2** 之后”，从而创建排序：

f1 f2 f3 f4

第三条命令随后替换文件 **f2** 和 **f3** 。由于文件 **f2** 和 **f3** 已存在于归档中，因此该命令序列无法完全替换为：

ar ra f1 newlib.a f2 f3

因为 **f2** 和 **f3** 在归档中的先前位置及相对顺序均被保留（无论命令行中指定文件的方式如何），所以生成以下归档：

newlib.a: f3' f2' f1 f4

警告

如果您是拥有适当权限的用户，则 **ar** 可以更改任意归档文件，即使该归档文件处于写保护状态也是如此。

如果同一文件在参数列表中被提及两次，则它会被放进归档两次。

如果归档中存在一个文件的多个副本，则 **ar** 会与该文件在归档中的第一个实例匹配。

ar 自动创建归档符号表，在 HP-UX 早期版本中，该任务是由 **ranlib** 执行的。 **z** 修饰符的作用是：禁止生成符号表，或在符号表已存在时使其无效。如果使用 **z** 修饰符构建了归档，则可以使用 **ranlib** 命令重建符号表。

文件

/var/tmp/ar* 临时文件

另请参阅

系统工具：

ld(1) 调用链接编辑器

其他：

acl(5) 访问控制列表
a.out(4) 汇编程序、编译程序和链接程序输出
ar(4) 归档格式
lorder(1) 查找对象文件或归档库的排序关系
ranlib(1) 重新生成归档符号表
strip(1) 从对象文件中去除符号和行号信息
tmpnam(3S) 为临时文件创建名称

文本和教程：

《HP-UX Linker and Libraries Online User Guide》
（请参阅 **+help** 选项）
《HP-UX Linker and Libraries User's Guide》
（有关排序的信息，请参阅 *manuals*(5)）

符合的标准

ar：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

as - 汇编程序

概要

备注

对于基于 Itanium® 的系统，请参阅 *as_ia*(1) 。

对于 PA-RISC 系统，请参阅 *as_pa*(1) 。

使用 **uname** 命令确定您的系统类型。在基于 Itanium 的系统上， **uname -m** 返回 **ia64** 。所有其他值表示 PA-RISC 系统。

另请参阅

as_ia(1)、 as_pa(1)、 uname(1)。

名称

asa - 解释 ASA 回车控制符

概要

asa [*files*]

说明

asa 用于解释 FORTRAN 程序的输出，该 FORTRAN 程序使用 ASA 回车控制符。它可处理在参数中指定了名称的文件（ *files* ），如果指定了 **-** 或未给定该文件名，则可处理标准输入。每行的第一个字符假定为控制符。以下控制符按照所述的方式进行解释：

（空白）
在打印之前输出单个换行符。

（空格）
（仅适用于 XPG4）。输出该行的剩余部分而不对其进行更改。

0 先输出 <换行符>，然后输出输入行的剩余部分。

1 在打印之前输出换页符。

+ 叠印上一行。

+ （仅适用于 XPG4）。上一行的 <换行符> 应替换为一个或多个由实现定义的字符，它们导致打印返回到列位置 1，其后是输入行的剩余部分。如果 + 是输入中的第一个字符，则应该与 <空格> 具有相同的效果。

以上述字符之外的字符开头的行将与以空白字符开头的行进行相同的处理。行的第一个字符 未打印。如果出现任何这样的行，则将相应的诊断消息发送标准错误。该程序强制每个输入文件的第一行在新的一页上开始。

（仅适用于 XPG4）。在遇到以上列出的字符之外的任意字符作为行的第一个字符时， asa 实用程序具有未指定的操作。

要查看使用 ASA 回车控制符并将其以常规形式显示的 FORTRAN 程序的输出，可以将 **asa** 用作过滤器：

a.out | asa | lp

然后，经过适当的格式设置和分页，输出将定向到行式打印机。可以使用以下命令在用户终端屏幕上查看之前发送到文件的 FORTRAN 输出：

asa *file*

外部语言环境影响

环境变量

LC_CTYPE 用于确定将文件内的文本解释为单字节和（或）多字节字符。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或 **LC_MESSAGES** 或将其设置为空字符串，则 **LANG** 的值将用作每个未指定

变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5)）而非 **LANG** 。

如果任一国际化变量中包含无效设置，则 **asa** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

另请参阅

efl(1)、 f77(1)、 fsplit(1)、 ratfor(1)。

符合的标准

asa：XPG4、POSIX.2

名称

as_ia: as - 在基于 Itanium 的系统中使用的汇编程序

概要

as [**-eVw**] [**+A32**] [**+A64**] [**+E32**] [**-elf32**] [**-elf64**] **-H** *lvl*[**=***what*[**,***what*]...] [**-o** *outfile*] **-W** *num*[**,***num*]... **-We** *num*[**,***num*]...
[*file*]

说明

as 可汇编指定的源文件 *file* ，或者在未指定 *file* 时汇编标准输入。汇编程序的输出是一个 ELF 浮动对象文件，这个文件必须先由 **ld** 处理，然后才能执行。

汇编程序的输出存储在 *outfile* 中。如果未指定 **-o** *outfile* 选项，汇编程序将自行构建缺省名称。如果未指定任何源文件， *outfile* 将为 **a.out** ；否则，源文件名的 **.s** 后缀（如果存在）会被去除，用 **.o** 后缀来替代，作为输出文件名。所有目录名都会从该名称中删除，以确保对象文件总是被写入当前目录中。

as 不执行任何宏处理。如果通过 C 编译程序调用了汇编程序，可以使用标准的 C 预处理器结构。

选项

as 可识别下列选项：

+A32 指定源文件包含 32 位 ABI 目标代码。源文件中的汇编程序指令 **.psr abi64** 会覆盖该选项。缺省情况下，对象文件是 32 位的 ELF 文件。

+A64 指定源文件包含 64 位 ABI 目标代码。源文件中的汇编程序指令 **.psr abi32** 会覆盖该选项。缺省情况下，对象文件是 64 位的 ELF 文件。

-e 允许在中止汇编过程前，有无限的容错量。缺省情况下，在汇编程序异常中止之前允许有一百个错误。

+E32 指定对象文件应为 32 位的 ELF 文件。这是缺省设置（另请参阅 **+A32** ）。请注意,将 64 位 ABI 目标代码写入 32 位 ELF 文件是有效的。对象文件中的所有 32 位地址都会在加载时通过添加零来扩展到 64 位。但这种零扩展方法可能会导致某些反向寻址无效（例如重定位）。

+E64 指定对象文件应为 64 位的 ELF 文件（请参阅 **+A64** ）。

-elf32 请参阅 **+E32** 。

-elf64 请参阅 **+E64** 。

-H *lvl*[**=***what*[**,***what*]...]
设置相关性检查。 *lvl* 只能为下列值之一： **off** 、 **warn** 或 **err** 。 *what* 只能为下列值之一： **data** 、 **impl** 或 **inst** 。

-o *outfile* 生成名为 *outfile* 的输出对象文件，而不构建一个缺省名。

-V 在汇编输入内容之前，将版本信息输出到标准错误中。

-W *num*[**,***num*]... 不输出指定的警告消息。

-We *num*[**,***num*]... 将指定的警告消息升级为错误。

-w 不输出所有警告消息。

外部语言环境影响

环境变量

NLSPATH 可确定消息清单的位置，以便处理 **LC_MESSAGES**。

SDKROOT 可控制需要调用哪个汇编程序，并支持多个跨平台开发工具包。 **SDKROOT** 变量指向特定 SDK 的根。对于如何验证变量值或调用的汇编程序是否适当，没有任何标准和规定。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

警告

汇编程序不会检查相关性。

诊断信息

当发生语法或语义错误时，标准错误中会显示单行诊断信息，还将显示发生错误的行号及文件名。

文件

/usr/lib/nls/C/as.cat	汇编程序错误消息清单
a.out	缺省汇编程序输出文件

另请参阅

cc(1)、 ld(1)、 elf(3E)。

名称

as_pa: as - PA-RISC 系统的汇编程序

概要

as [**-eflsuV**] [**-o** *objfile*] [**-p** *number*] [**-v** *xrfile*] [**-w**[*number*]]... [**+DA***architecture*] [**+z**] [**+Z**] [*file*]...

说明

as 命令从文件或标准输入中汇编源文本，并生成适用于链接编辑器 **ld** （请参阅 *ld*(1) ）的浮动对象文件。

仅当未给出 *file* 参数时，才从标准输入中读取源文本。标准输入不能是设备文件，例如终端。 *option* 和 *file* 参数可以在命令行上混合使用。每个指定的 *option* 应用于每个指定的 *file* 或标准输入。源文件将连接起来形成一个输入流。

如果未指定 **-o** *objfile* 选项，则 **.s** 后缀（如果有）将从最后一个源文件名称的末尾去除，并将 **.o** 添加到该名称中以形成缺省对象代码输出文件的文件名。

as 输出不是最优化的。 **as** 可创建浮动对象文件，该文件必须先由 **ld** 进行处理，然后才能成功执行（请参阅 *ld*(1)）。

cc 编译程序通常运行 C 预处理器 **cpp** （请参阅 *cpp*(1) ），然后调用 **as** 来汇编 **.s** 文件和 **/usr/lib/pcc_prefix.s** ，随后再调用 **ld** 。

选项和操作数

as 可识别下列选项和操作数：

file	包含汇编程序源代码的文本文件。
-e	在放弃汇编进程之前允许有数量不限的错误。缺省情况下，在汇编程序异常中止之前允许有 100 个错误。
-f	将 **.BALLINFO** 指令的缺省值设置为 **CALLS** ，通常情况下，省略了 **CALLER** 、**CALLS** 或 **NO_CALLS** 参数的 **.BALLINFO** 的缺省值为 **NO_CALLS** 。
-l	汇编之后，将程序列表写入标准输出中。该列表显示指令的偏移量以及字段的实际值。
-o *objfile*	命名输出对象文件 *objfile* ，而不使用指定的最后一个 *file* 的文件名的缺省 **.o** 后缀。
-p *number*	将 **.EXPORT** 指令的缺省权限级别设置为 *number* 。缺省情况下，以权限级别 3 导出所有用户级别过程。
-s	将输出文件的后缀设置为 **.ss** 而不是 **.o** 。文件将具有一个适于转换为 ROM 烧制程序的格式。
-u	不要创建展开描述符。为避免对 **.BALLINFO** 指令的需要，不得使用 **.ENTER** 和 **.LEAVE** 指令。
-v *xrfile*	将交叉引用数据写入名为 *xrfile* 的文件中。

-V 在汇编源文本之前，将汇编程序的版本号输出到标准错误中。

-w[*number*] 如果没有提供 *number* ，则禁止显示所有警告消息，或者，仅禁止显示提供了 *number* 的警告。可以使用多个 **-w***number* 选项，以禁止显示额外的警告消息。

+DA *architecture* 为指定的 *architecture* 汇编代码。不赞成使用该选项。推荐的选择 *architecture* 的方法是，使 **.LEVEL** 指令包含在汇编源文件中。

汇编程序使用下列优先顺序确定目标体系结构。

1. 使用汇编源文件中的 **.LEVEL** 指令。
2. 使用 **+DA** 命令行规范。
3. 使用缺省体系结构 **PA_RISC_1.0** 。

+z,+Z 这两个选项均用于构建共享库。有关这些选项的完整论述，请参阅《HP-UX Linker and Libraries User’s Guide》手册。

外部语言环境影响

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

如果发生句法或语义错误，则一行诊断信息将写入标准错误中，其中包括文件名和发生错误的行号。格式如下：

as: "*filename*",line *line*: error *error*: *message*
source = *source-line*

警告

as 不调用 *cpp*(1) 或 *m4*(1) 来执行宏处理。

文件

/usr/include/hard_reg.h	硬件寄存器定义
/usr/include/soft_reg.h	软件调用约定寄存器定义
/usr/include/std_space.h	标准空间和子空间定义
/usr/lib/nls/msg/C/as.cat	汇编程序错误消息清单
/usr/lib/pcc_prefix.s	空间、子空间和寄存器定义
*file***.o**	对象文件

另请参阅

adb(1)、 cc_bundled(1)、 cpp(1)、 ld(1)、 nm(1)、 crt0(3)。

《Assembly Language Reference Manual》、

《Precision Architecture and Instruction Reference Manual》、

《Procedure Calling Conventions Reference Manual》。

名称

at、batch - 立即或稍后执行批处理命令

概要

从标准输入中输入在指定时间运行的命令:

at [**-m**] [**-q** *queue*] **-t** *spectime*
commands
eof

at [**-m**] [**-q** *queue*] *time* [*date*] [**next** *timeunit* | **+***count timeunit*]
commands
eof

从文件中输入在指定时间运行的命令:

at -f *job-file* [**-m**] [**-q** *queue*] **-t** *spectime*

at -f *job-file* [**-m**] [**-q** *queue*] *time* [*date*] [**next** *timeunit* | **+***count timeunit*]

列出调度的作业:

at -d *job-id* ...

at -l [*job-id* ...]

at -l -q *queue*

取消（删除）调度的作业:

at -r *job-id* ...

从标准输入中输入作为批处理进程运行的命令:

batch
commands
eof

从文件中输入作为批处理进程运行的命令:

batch < *job-file*

说明

at 和 **batch** 命令用于调度 **cron** 守护程序所执行的作业（请参阅 *cron*(1M) ）。

at 将作业安排在指定时间执行。 **at** 也可列出 (**-l**) 或删除 (**-r**) 现有的已调度 **at** 和 **batch** 作业。

batch 安排作业立即执行，或者在系统负载程度允许的情况下执行。

可以通过以下方式之一在作业中输入命令:

- 通过键盘在紧接 **at** 或 **batch** 命令行之后的单独行上，后接当前定义的 *eof* （文件结束标志）字符以终止输入。缺省的 *eof* 是 Ctrl-**D** 。可以在您的环境中将其重新定义（请参阅 *stty*(1) ）。

- 使用 **at** 命令的 **-f** 选项从脚本文件中读取输入。
- 从前面的命令通过管道发送的输出中。

选项和参数

at 可识别下列选项和参数。

commands 可以由 **at** 或 **batch** 作为 Shell 脚本执行的一个或多个 HP-UX 命令。

eof 文件结束标志字符。除非您的环境中另行定义，否则缺省值为 Ctrl-**D** 。

job-file 现有文件的路径名。

job-id 在最初调度作业时由 **at** 或 **batch** 报告的作业标识符。

-d *job-id* ... 显示指定作业的内容。无特权的用户仅限于显示有关该用户拥有的作业的信息。具有适当特权的用户可以显示有关所有作业的信息。

-f *job-file* 读取 *job-file* 中包含的命令，而不是使用标准输入。

-l [*job-id* ...] 列出指定的作业。如果未给定 *job-id* ，则列出所有作业。

-m 在作业运行完后向调用的用户发送邮件，声明作业的完成。除非在作业中重定向到他处，否则作业生成的标准输出和标准错误也将自动通过邮件发送给用户。

-q *queue* 将指定作业发送到指定的 *queue* （请参阅 *queuedefs*(4) ）。可以使用从 **a** 、 **b** 和 **d** 到 **y** 的队列。 **at** 在缺省情况下将使用队列 **a** 。 **batch** 始终使用队列 **b** 。除 **b** 之外的所有队列均需要指定 *time* 或 **-t** 。 **at -qb** 等效于 **batch** 。与 **-l** 选项一起使用时，将搜索限制到特定的队列。

-r *job-id* ... 删除由各个 *job-id* 指定的作业。

-t *spectime* 定义开始作业的绝对时间。

spectime 以下格式的日期和时间：

[[*CC*] *YY*] *MMDDhhmm* [**.** *ss*]

其中的十进制数位对如下：

CC	年份的前两位数字（ 19、20 ）。
YY	年份的下两位数字（ 69-99 、 **00-68** ）。请参阅“警告”。
MM	一年的月份 (**01-12**) 。
DD	一月中的日期 (**01-31**) 。
hh	一天中的小时 (**00-23**) 。
mm	一小时中的分钟 (**00-59**) 。
ss	一分钟中的秒 (**00-61**) 。

如果省略 *CC* 和 *YY* ，则缺省值是当前年份。

如果省略 *CC* 而 *YY* 介于 **69-99** 的范围内，则 *CC* 缺省为 **19** 。否则，**CC** 缺省为 **20** 。

ss 的范围支持两闰秒。如果 *ss* 是 **60** 或者 **61** 并且受 **TZ** 环境变量影响的所得时间不使用闰秒作单位，则时间将设置为 *mm* 之后的整分钟。

如果省略 *ss* ，则缺省为 **00** 。

time [*date*]

定义开始作业的基准时间。

time

指定为一、二或四位数字的时间。一位和二位数字表示小时；四位数字表示小时和分钟。

或者， *time* 也可以指定为两个由冒号 (**:**) 、单引号 (**'**) 、字母 *h* (**h**) 、句点 (**.**) 或逗点 (**,**) 分隔的数字。分隔符和表示分钟的数字之间可以存在空格。如果在 *langinfo*(5) 中定义，则可以使用特殊时间单位字符。

可以追加 **am** 或 **pm** 来指示上午或下午。否则，将按 24 小时制时钟计时。例如， **0815** 、 **8:15** 、 **8'15** 、 **8h15** 、 **8.15** 和 **8,15** 均读作上午八点过 15 分。可以使用后缀 **zulu** 和 **utc** 来指定国际标准时间 (UTC)，它等效于格林威治标准时间 (GMT)。

此外还识别特殊名称 **midnight** 、 **noon** 和 **now** 。

date

一周的星期几（全写或缩写）或者由日、月和（可选）年组成的日期。日期和年份字段必须是数值，月份可以是全写名、简写名或数值。日期字符串中的字段由以下标点符号进行分隔：斜线 (**/**) 、连字符 (**-**) 、句点 (**.**) 和逗点 (**,**) 。如果在 *langinfo*(5) 中定义，则可以使用特殊日期单位字符。其值大于 31 的字段将作为年份字段进行处理，日期字符串中的剩余两个字段则作为月份和日期字段进行处理。否则，如果给定的日期具有歧义（如 **2/5** 或 **2/5/10** ），则使用 **D_T_FMT** 字符串（如果在 *langinfo*(5) 中定义）来解决歧义。

此外还识别两种特殊的日期 **today** 和 **tomorrow** 。如果没有给定 *date*，则假定为 **today** （如果给定时间大于当前时间）或 **tomorrow** （如果给定时间小于当前时间）。

如果给定月份小于当前月份（并且未给定年份），则假定为下一年。处于范围 69 到 99 的两位数字年份将扩展为 1969 到 1999；处于范围 00 到 68 的两位数字年份将扩展为 2000 到 2068。

next *timeunit* | **+** *count timeunit*

在由 *time* [*date*] 指定的基准时间后将执行日期和时间延迟指定数量的时间单位。

count

十进制数字。 **next** 等效于 **+1** 。

timeunit 以下时间单位之一： **minutes** 、 **hours** 、 **days** 、 **weeks** 、 **months** 或 **years** （或其单数形式）。

处理作业的方式

接受作业时， **at** 和 **batch** 将以如下形式向标准错误输出一条消息：

job *job-id* **at** *execution-date*

其中 *job-id* 是 *jobnumber***.***queue* 形式的作业标识符（如 **756284400.a** ）， *execution-date* 是释放作业以执行的日期和时间。

如果您的登录 Shell 不是 POSIX Shell (**/usr/bin/sh**)，该命令还会输出一条警告消息：

warning: commands will be executed using /usr/bin/sh

at 作业缺省为队列 **a** 。 **batch** 作业始终会进入队列 **b** 。请参阅 **-q** 选项。

at 或 **batch** 作业由 **/var/spool/cron/atjobs** 中存储的脚本组成，该脚本包括两个部分，可以由 POSIX Shell 执行。

第一个部分设置环境，在您发出 **at** 或 **batch** 命令时对环境进行匹配处理。它包括当前 Shell 环境变量、当前目录、 **umask** 和 **ulimit** （请参阅 *ulimit*(2) 、 *umask*(1) 和 *proto*(4) ）。将丢失打开的文件描述符、陷阱和优先级。

第二个部分由您输入的命令组成。

当 **cron** 调度进程时，将启动 POSIX Shell 来执行脚本。

在任何时候从队列中执行作业数通过文件 **/var/adm/cron/queuedefs** 中的参数来控制（请参阅 *queuedefs*(4) ）。

来自作业的标准输出和标准输入将通过邮件发送给用户（除非它们在作业中重定向到他处）。

调度的作业不受 **SIGHUP** 挂起信号的影响，则用户注销后仍保持调度状态。

如果用户的用户名出现在文件 **/usr/lib/cron/at.allow** 中，他们就可以使用 **at** 和 **batch** 命令。如果不存在该文件，只要用户的用户名 不出现在文件 **/usr/lib/cron/at.deny** 中，用户就可以使用 **at** 和 **batch** 。如果这两个文件均不存在，则只有超级用户才能提交作业。如果只存在 **at.deny** 但该文件为空，则所有用户均可使用 **at** 和 **batch** 。 **allow/deny** 文件在每一行包含一个用户名。

所有用户均可列出和删除其自己的作业。拥有适当特权的用户可以列出和删除他人的作业。

外部语言环境影响

环境变量

LC_TIME 确定日期和时间字符串的格式和内容。

LC_MESSAGES 用于确定显示消息的语言。

LC_MESSAGES 也确定还可使用哪种语言指定单词 **days** 、 **hours** 、 **midnight** 、 **minutes** 、 **months** 、 **next** 、 **noon** 、 **now** 、 **today** 、 **tomorrow** 、 **weeks** 、 **years** 及其单数形式。

如果未在环境中指定 **LC_TIME** 或 **LC_MESSAGES** 或将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省的“C”（请参阅 *lang*(5) ）

而非 **LANG** 。

如果任一国际化变量中包含无效设置，则所有国际化变量都会缺省为“C”（请参阅 *environ*(5)）。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

注释

batch 命令将为其调度的每个批处理作业请求一个唯一的 *job-id* 。请求唯一 *job-id* 的最大尝试次数限制在 100。如果在 100 次尝试后仍未成功， **batch** 命令将退出并发出消息 **queue full** 。如果已安装 **BatchConfig** 产品，则可以通过设置 **/etc/default/cron** 文件中的变量 **BATCH_MAXTRYS=***value* 来配置该数字。 **BATCH_MAXTRYS** 的值可以是任意正数或者是字符串 **INFINITE** （缺省值）。如果该值设置为 **INFINITE** ， **batch** 将不断请求唯一的 *job-id* ，直到成功收到该作业 ID 为止。

返回值

退出代码设置为以下代码之一：

0 成功完成

1 失败

诊断信息

at 将为语法错误和超出范围的时间生成自述性消息。

warning: commands will be executed using /usr/bin/sh

如果您的登录 Shell 不是 POSIX Shell (**/usr/bin/sh**)，则 **at** 和 **batch** 将生成一条警告消息，提示 **at** 和 **batch** 作业正在使用 **/usr/bin/sh** 执行。

举例

以下命令显示在当前时间的五分钟后运行名为 **delayed-job** 的 POSIX Shell 脚本文件的三种不同方式：

```
at -f delayed-job now + 5 minutes
cat delayed-job | at now + 5 minutes
at now + 5 minutes <delayed-job
```

在系统负载程度允许的情况下运行典型的 HP-UX 命令（本例中是 nroff），并将标准输出和标准错误重定向到文件：

```
batch
nroff source-file >output-file 2>error-file
eof     （缺省值是 Ctrl-D）
```

在 2013 年 12 月 27 日上午 12:20 运行主目录中的 **future** 所包含的作业：

```
at -f $HOME/future -t201312271220.00
```

将标准错误重定向到管道（在 Shell 过程中非常有用）。请注意，输出重定向指定的顺序非常重要。标准错误重

定向到标准输出的目标位置；标准输出重定向到文件；初始“标准输出”（现在包括原来的标准错误）通过管道发送到 **mail** 程序。

batch <<!! (sets *eof* temporarily to **!!**)
nroff input-file 2>&1 1> output-file | mail *loginid*
!!

在下周星期二上午 5:00 运行主目录中的 **jobfile** 所包含的作业：

at -f $HOME/jobfile 5am tuesday next week

下周星期二的一周后的上午 5:00 运行相同的作业（即提前 2 个星期二进行安排）：

at -f $HOME/jobfile 5am tuesday + 2 weeks

将一个命令添加到主目录内目录 **jobs** 中名为 **weekly-run** 的文件，使其在每次运行时自动对其自己进行重新调度。该示例在 1900 年每周星期四的下午 7:00 进行重新调度：

echo "sh $HOME/jobs/weekly-run" | at 1900 thursday next week

以下命令显示 **at** 可识别的几种形式，并包括本地语言用法：

at 0815 Jan 24
at 8:15 Jan 24
at 9:30am tomorrow
at now + 1 day
at -f job 5 pm Friday
at 17:40 Tor. **# in Danish**
at 17h46 demain **# in French**
at 5:30 26. Feb. 1988 **# in German**
at 12:00 26-02 **# in Finnish**

警告

如果 *date* 参数以数字开头并且 *time* 参数也是不带后缀的数值，则 *time* 参数应该是一个四位数字，以便正确地解释为小时和分钟。

如果在一个 **at** 命令中同时使用 **next** 和 **+***count*，则将接受第一个运算符，并以静默方式忽略尾随的运算符。

如果在同一个命令中同时使用 **-t** 和 *time* ...，则将接受第一个运算符，并以静默方式忽略第二个运算符。

如果用于与 **cron** 通信的 FIFO 已满， **at** 将暂停，直到 **cron** 从 FIFO 中读取足够的消息，给 **at** 尝试撰写的消息留出空间。如果 **at** 撰写消息的速度比 **cron** 的处理速度更快，或者 **cron** 未执行，则可能出现这种情况。

调度的进程在后台运行。任何调用其自身的脚本文件都将导致用户或系统耗尽可用的进程。

如果作业的执行时请求复制当前调度作业的执行时间，则新的作业时间将设置为下一可用秒。

at 将不调度其开始时间早于当前时代（国际标准时间 1970 年 1 月 1 日 00:00:00）的作业。 **at** 不会将作业调度到 2037 年之后。

相关内容

HP Process Resource Manager

如果安装并配置了可选的 HP Process Resource Management (PRM) 软件，作业将在调度该作业的用户的初始进程资源组中启动。作业的初始组在作业启动时（而不是在作业调度时）确定。如果未定义用户的初始组，作业将在用户缺省组 (**PRMID=1**) 中运行。有关如何配置 HP PRM 的说明，请参阅 *prmconfig*(1)；有关如何确定用户初始进程资源组的说明，请参阅 *prmconf*(4)。

作者

at 由 AT&T 和 HP 联合开发。

文件

/usr/bin/sh	POSIX Shell
/var/adm/cron	主 **cron** 目录
/var/adm/cron/.proto	该文件包含添加到 **at** 作业文件中，使 **at** 作业的环境与当前环境相同的 Shell 命令集。请参阅 *proto(4)*。
/usr/lib/cron/at.allow	允许用户的列表
/usr/lib/cron/at.deny	拒绝用户的列表
/var/adm/cron/queuedefs	调度信息
/var/spool/cron/atjobs	假脱机区域

另请参阅

crontab(1)、kill(1)、mail(1)、nice(1)、ps(1)、sh(1)、stty(1)、cron(1M)、proto(4)、queuedefs(4)。

HP Process Resource Manager：《HP Process Resource Manager User's Guide》中的 prmconfig(1)、prmconf(4)

符合的标准

at：SVID2、SVID3、XPG2、XPG3、XPG4

batch：SVID2、SVID3、XPG2、XPG3、XPG4

名称

attributes - 描述一个音频文件

概要

/opt/audio/bin/attributes *filename*

说明

本命令提供关于某个音频文件的各种信息，包括文件格式、数据格式、采样率、通道数、数据长度和头长度。

举例

下面是一个对 HP-UX 提供的音频文件使用 **attributes** 的示例。

$ /opt/audio/bin/attributes /opt/audio/sounds/welcome.au

File Name: /opt/audio/sounds/welcome.au
File Type: NeXT/Sun
Data Format: Mu-law
Sampling Rate: 22050
Channels: Mono
Duration: 1.972 seconds
Bits per Sample: 8
Header Length: 40 bytes
Data Length: 43492 bytes

作者

attributes 由 HP 开发。

Sun 是 Sun MicroSystems, Inc. 的商标。

NeXT 是 NeXT Computers, Inc. 的商标。

另请参阅

audio(5)、 asecure(1M)、 aserver(1M)、 convert(1)、 send_sound(1)。

《Using Audio Developer’s Kit》

名称

awk - 依据模式匹配条件扫描和处理的语言

概要

awk [**-F***fs*] [**-v** *var*=*value*] [*program* | {**-f** *progfile*}...] [*file*]...

说明

awk 扫描每个输入文件 *file* ，查找与 *program* 中或者指定为 **-f** *progfile* 的一个或多个文件中指定的一组模式中的任一模式相匹配的行。对于每个模式，当 *file* 中的某行与该模式相匹配时将执行一个关联的操作。将对照每个模式操作语句的模式部分匹配每一行，并对每个匹配的模式都执行关联操作。文件名 **-** 表示标准输入。将形式为 *var*=*value* 的所有 *file* 都视为一种赋值而不是文件名。如果它是文件名，则在打开它时对赋值进行求值，除非使用了 **-v** 选项。

输入行由字段组成，各字段由空格或正则表达式 **FS** 分隔。字段由 **$1** 、 **$2** ... 表示； **$0** 指代整行。

选项

awk 识别以下选项和参数：

-F *fs* 指定用于分隔字段的正则表达式。缺省情况下，识别空格和制表符，并忽略前导空格和多个制表符。如果使用 **-F** 选项，则不再忽略前导输入字段分隔符。

-f *progfile* 指定 awk 程序文件。最多可以指定 100 个程序文件。这些文件中模式操作语句的执行顺序与文件的指定顺序相同。

-v *var*=*value* 使 *var*=*value* 赋值在执行 **BEGIN** 操作（如果它存在）之前发生。

语句

模式操作语句的格式如下：

pattern **{** *action* **}**

如果缺少 **{** *action* **}** ，则表示打印行；如果缺少模式，则始终匹配。模式操作语句由换行符或分号分隔。

操作是一组语句。语句可以是以下其中一项：

```
if(expression) statement [ else statement ]
while(expression) statement
for(expression;expression;expression) statement
for(var in array) statement
do statement while(expression)
break
continue
{[statement]...}
expression        # commonly  var = expression
print [expression-list] [ > expression]
printf format [, expression-list] [ > expression]
```

return [*expression*]

next # skip remaining patterns on this input line.

delete *array* [*expression*] # delete an array element.

exit [*expression*] # exit immediately; status is *expression*.

语句以分号、换行符或右大括号结尾。空的 *expression-list* 代表 **$0** 。字符串常量用引号括起来 (**""**)，其内可识别常见的 C 转义符。根据需要表达式可以是字符串值或数值，由运算符 **+** 、 **-** 、 ***** 、 **/** 、 **%** 、 **^** （求幂）和连接符（指空白符）组成。运算符 **++** 、 **--** 、 **+=** 、 **-=** 、 ***=** 、 **/=** 、 **%=** 、 **^=** 、 ****=** 、 **>** 、 **>=** 、 **<** 、 **<=** 、 **==** 、 **!=** 、 **""** （双引号、字符串转换运算符）和 **?:** 也可供表达式使用。变量可以是标量、数组元素（表示为 *x*[*i*]）或字段。变量被初始化为空字符串。数组下标可以是任何字符串，不一定是数值（这样就允许使用关联内存形式）。允许使用多个下标，如 **[***i***,***j***,***k***]**。各成分连接在一起，由 **SUBSEP** 的值分隔。

print 语句在标准输出上（如果存在 **>***file* 或 **>>***file* ，则在文件上；如果存在 |*cmd* ，则在管道上）打印其参数，各参数由当前的输出字段分隔符分隔，并以输出记录分隔符结尾。 *file* 和 *cmd* 可以是文字名称或括在括号中的表达式。不同语句中的相同字符串值表示同一打开文件。 **printf** 语句根据其格式（请参阅 *printf*(3) ）来设置其表达式列表的形式。

内置函数

内置函数 **close(***expr***)** 负责关闭由 **print** 或 **printf** 语句打开的文件或管道 *expr* ，或关闭使用同一字符串值 *expr* 的 **getline** 命令调用。如果成功，则此函数返回零；否则，它返回非零。

常用的函数 **exp** 、 **log** 、 **sqrt** 、 **sin** 、 **cos** 、 **atan2** 都是内置函数。其他内置函数有：

blength[([*s*])] 它的关联参数（视为字符串）的长度（以字节为单位），如果没有参数，则采用 **$0** 的长度。

length[([*s*])] 它的关联参数（视为字符串）的长度（以字符为单位），如果没有参数，则采用 **$0** 的长度。

rand() 返回 0 与 1 之间的随机数。

srand([*expr*]**)** 为 *rand* 设置种子值，并返回以前的种子值。如果未指定参数，则将一天中的时间用作种子值；否则使用 *expr* 。

int(*x***)** 截断为整数值

substr(*s***,** *m* [**,** *n*]**)** 返回 *s* 的子字符串，它最多包含 *n* 个字符，从位置 *m* （从 1 开始编号）开始计数。如果省略 *n* ，则子字符串最多不能超过字符串 *s* 的长度。

index(*s***,** *t***)** 返回字符串 *s* 中字符串 *t* 首次出现的位置（以字符为单位、从 1 开始编号），或者如果 *t* 不存在则返回零。

match(*s***,** *ere***)** 返回字符串 *s* 中扩展的正则表达式 *ere* 出现的位置（以字符为单位、从 1 开始编号），或者如果 *ere* 不存在则返回 0。将变量 **RSTART** 和 **RLENGTH** 设置为匹配字符串的位置和长度。

split(*s***,** *a***[,** *fs***])** 将字符串 *s* 拆分为数组元素 *a***[1]**、 *a***[2]**、...、 *a***[***n***]**，并返回 *n* 。分隔是通过正则表达式 *fs* 或字段分隔符 **FS** （如果未指定 *fs* ）完成的。

sub(*ere***,** *repl* **[,** *in***])** 用 *repl* 替换字符串 *in* 中出现的第一个扩展的正则表达式 *ere* 。如果未指定 *in* ，则使用 **$0** 。

gsub 与 **sub** 相同，只不过所有出现的正则表达式均被替换； **sub** 和 **gsub** 返回替换的数量。

sprintf(*fmt***,** *expr***, ...)** 设置根据 *printf*(3S) 格式 *fmt* 对 *expr*... 进行格式设置而生成的字符串

system(*cmd***)** 执行 *cmd* 并返回其退出状态

toupper(*s***)** 将参数字符串 *s* 转换为大写并返回结果。

tolower(*s***)** 将参数字符串 *s* 转换为小写并返回结果。

内置函数 **getline** 将 **$0** 设置为当前输入文件中的下一条输入记录； **getline <** *file* 将 **$0** 设置为 *file* 中的下一条记录。而 **getline** *x* 设置变量 *x* 。最后， *cmd* **| getline** 将 *cmd* 的输出传输到 **getline** 中； **getline** 的每个调用都返回 *cmd* 输出中的下一行。在所有情况下， **getline** 返回 1 表示成功输入，返回 0 表示文件结束，返回 -1 表示错误。

模式

模式是正则表达式和关系表达式的任意布尔组合（带有 **! || &&** ）。 **awk** 支持扩展的正则表达式，如 *regexp*(5) 中所述。模式中的独立的正则表达式适用于整行。正则表达式也可以出现在关系表达式中，使用的是运算符 ~ 和 **!~**。 */re/* 是常量正则表达式；任何字符串（常量或变量）都可用作正则表达式，但除了在模式中的独立的正则表达式的位置。

一个完整模式可由用逗号分隔的两个模式组成；在这种情况下，即使出现第二个模式，仍然从出现第一个模式的地方对所有行执行操作。

关系表达式为以下其中一种表达式：

expression matchop regular-expression
expression relop expression
expression **in** *array-name*
(*expr,expr,...***)** *in* **array-name**

其中 *relop* 是 C 中六个关系运算符中的任何一个， *matchop* 是 ~ （匹配）或 **!~** （不匹配）。条件表达式是算术表达式、关系表达式或这两者的布尔组合。

通常在第一个输入行之前和最后一个输入行之后，使用特殊模式标记 **BEGIN** 和 **END** 来捕获读取的控制范围。 **BEGIN** 和 **END** 不与其他模式进行组合。

特殊字符

awk 可识别正则表达式和字符串中的下列特殊转义序列：

转义符	含义
\a	报警符

\b	退格符
\f	换页符
\n	换行符
\r	回车符
\t	制表符
\v	纵向制表符
*****nnn*	1 到 3 位八进制值 *nnn*
\x*hhh*	1 到 n 位十六进制数

变量名称

具有特殊含义的变量名称是:

FS 输入字段分隔符正则表达式；缺省情况下为空格；也可通过选项 **-F** *fs* 进行设置。

NF 当前记录中的字段数。

NR 当前记录的序号，从输入的开头算起。在 **BEGIN** 操作内，该值为零。在 **END** 操作内，该值为处理的最后一条记录的编号。

FNR 当前记录在当前文件中的序号。在 **BEGIN** 操作内，该值为零。在 **END** 操作内，该值为在最后一个文件中处理的最后一条记录的编号。

FILENAME 当前输入文件的路径名。

RS 输入记录分隔符；缺省情况下为换行符。

OFS **print** 语句的输出字段分隔符；缺省情况下为空格。

ORS **print** 语句的输出记录分隔符；缺省情况下为换行符。

OFMT 数字的输出格式（缺省为 **%.6g** ）。如果 **OFMT** 的值未规定为浮点格式，则结果也不做规定。

CONVFMT 数值的内部转换格式（缺省为 **%.6g** ）。如果 **CONVFMT** 的值未规定为浮点格式，则结果也不做规定。

有关 **CONVFMT** 的其他信息，请参考外部语言环境影响下的 **UNIX95** 变量。

SUBSEP 多维数组的下标分隔符字符串；缺省值为“\034”

ARGC **ARGV** 数组中的元素个数。

ARGV 命令行参数的数组，不包括选项和编号为从零到 **ARGC**-1 的 *program* 参数。

可以修改或添加 **ARGV** 中的参数；可以更改 **ARGC** 。在每个输入文件结束时，**awk** 会将 **ARGV** 的下一个非空元素（直到 **ARGC**-1 的当前值，包括该值）视为下一个输入文件的名称。这样，将 **ARGV** 的元素设置为空意味着不会将它视为输入文件。名称 **-** 代表标准输入。如果参数与 *assignment* 操作数的格式匹配，则将此参数视为赋值而不是 *file* 参数。

ENVIRON 环境变量的数组；下标为名称。例如，如果环境变量 **V=thing** ，则 **ENVIRON["V"]** 生成 **thing** 。

RSTART 与 **match** 函数匹配的字符串的起始位置，从 1 开始编号。此值总是等于 **match** 函数的返回值。

RLENGTH 与 **match** 函数匹配的字符串的长度。

可以按如下所示定义函数（在模式操作语句的位置）：

function foo(a, b, c) { ...; return x }

如果为标量，则按值传递参数；如果为数组名称，则按引用传递参数。可以递归地调用函数。参数对于函数是局部的；所有其他变量都是全局的。

请注意，如果在 HP-UX 命令行中将模式操作语句用作 **awk** 命令的参数，则必须将模式操作语句括在单引号中，以防止 Shell 执行它。例如，要打印长度超过 72 个字符的行，则在脚本 （**-f** *progfile* 命令格式）中使用的模式操作语句为:

length > 72

用作 **awk** 命令的参数的相同模式操作语句按以下方式括在引号中:

awk 'length > 72'

外部语言环境影响

环境变量

UNIX95 如果已定义，则指定将 XPG4 行为用于此命令。对 XPG4 的更改包括对上面指定的整个行为的支持，并包括以下行为更改:

- 如果未指定 **CONVFMT** 但设置了 **UNIX95**，则缺省情况下将 **%d** 用作内部数字转换格式。

LANG 为没有设置或设置为 null（空）的国际化变量提供了缺省值。如果 **LANG** 未设置或设置为 null（空），则会使用缺省值“C”（请参阅 *lang*(5)）。如果任一国际化变量中包含无效设置，则 **awk** 将按照所有国际化变量都设为“C”的条件执行。请参阅 *environ*(5) 。

LC_ALL 如果设为非空字符串值，则会覆盖所有其他国际化变量的值。

LC_CTYPE 用于确定作为单字节和（或）多字节字符的文本的解释、作为可打印字符的字符的分类，以及与正则表达式中字符类表达式匹配的字符。

LC_NUMERIC 用于确定在解释数字输入、执行数字值与字符串值之间的转换以及设置数字输出格式时所使用的小数分隔符。无论使用什么语言环境，句点字符（POSIX 语言环境的小数点字符）都是在处理 **awk** 程序（包括命令行参数中的赋值）时识别的小数点字符。

LC_COLLATE 用于确定正则表达式内范围、等效类和多字符排序元素的行为的语言环境。

LC_MESSAGES
所确定的语言环境作用于写入到标准错误的诊断消息和写入到标准输出的信息性消息的格式及内容。

NLSPATH 用于确定消息清单的位置，以便处理 **LC_MESSAGES**。

PATH 用于确定查找 **system(cmd)** 所执行的命令以及输入和输出管道时的搜索路径。

此外，通过设置 **awk** 的变量 **ENVIRON** ，所有环境变量都将为可见的。

国际代码集支持

支持单字节字符代码集和多字节字符代码集，只是变量名必须仅包含 ASCII 字符，正则表达式必须仅包含有效字符。

诊断信息

awk 对于每条记录最多支持 199 个参数字段（ $1 ， **$2** ……， **$199** ）。

举例

打印长度超过 72 个字符的行：

```
length > 72
```

按相反顺序打印前两个字段：

```
{ print $2, $1 }
```

同前，打印由逗号和（或）空白符和制表符分隔的输入字段：

```
BEGIN { FS = ",[ \t]*|[ \t]+" }
      { print $2, $1 }
```

对第一列进行合计，打印总和以及平均值：

```
        { s += $1 }"
END     { print "sum is", s, " average is", s/NR }
```

打印开始/停止对之间的所有行：

```
/start/, /stop/
```

模拟 **echo** 命令（请参阅 *echo*(1) ）：

```
BEGIN   {                       # Simulate echo(1)
        for (i = 1; i < ARGC; i++) printf "%s ", ARGV[i]
        printf "\n"
        exit }
```

作者

awk 由 AT&T、IBM、OSF 和 HP 开发。

另请参阅

lex(1)、 sed(1)。

A.V. Aho、 B.W. Kernighan、 P.J. Weinberger：《The AWK Programming Language》， Addison-Wesley， 1988。

符合的标准

awk： SVID2、 SVID3、 XPG2、 XPG3、 XPG4、 POSIX.2

名称

banner - 使用大尺寸字母输出横幅

概要

banner *strings*

说明

banner 在标准输出中以大尺寸字母输出其参数（每个参数最多包含 10 个字符）。

每个参数单独输出在一行上。请注意包含多个单词的参数必须用引号括住，只有这样它才能输出在同一行中。

举例

在屏幕上以大尺寸字母输出消息“Good luck Susan”：

banner "Good luck" Susan

单词 **Good luck** 显示在同一行中， **Susan** 显示在另一行中。

警告

X/Open 标准中可能会撤消此命令。使用此命令的应用程序可能无法移植到其他供应商的平台上。

另请参阅

echo(1)。

符合的标准

banner：SVID2、SVID3、XPG2、XPG3

名称

basename、dirname - 提取路径名的组成部分

概要

basename *string* [*suffix*]

dirname [*string*]

说明

basename 从 *string* 中删除所有以 **/** 结束的前缀并删除 *suffix* （如果在 *string* 中有的话），并将结果输出到标准输出中。如果 *string* 完全由斜线字符组成，则 *string* 将被设置为单个斜线字符。如果 *string* 中有结尾斜线字符，则这些字符会被删除。如果指定了后缀操作数，但是该后缀与 *string* 中剩余的字符不相同，却与 *string* 中剩余字符的某个后缀相同，则会从 *string* 中删除该后缀。 **basename** 通常用于 Shell 过程中的命令替换标记 (`...`) 中。

dirname 输出 *string* 中除最低级别的路径名之外的所有内容。如果 *string* 中不包含目录部分，则 **dirname** 将返回 **.**，表示当前工作目录。

外部语言环境影响

环境变量

LC_CTYPE 确定将字符串和后缀（对于 basename 命令）解释为单字节和（或）多字节字符。

如果未在环境中指定 **LC_CTYPE** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而不使用 **LANG** 。如果任一国际化变量包含无效设置， **basename** 和 **dirname** 将认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

下面用参数 **/usr/src/cmd/cat.c** 调用的 Shell 脚本将编译命名的文件，并将输出结果移动到当前目录中名为 **cat** 的文件中：

```
cc $1
mv a.out ‘basename $1 .c‘
```

以下示例将 Shell 变量 **NAME** 设置为 **/usr/src/cmd** ：

```
NAME=‘dirname /usr/src/cmd/cat.c‘
```

返回值

basename 和 **dirname** 返回下列值之一：

0 成功完成。

1 命令行参数数量有误。

另请参阅

expr(1)、sh(1)。

符合的标准

basename：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

dirname：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

bc - 任意精度的算术语言

概要

bc [**-c**] [**-l**] [*file* ...]

说明

对类似于 C 语言的语言， **bc** 是一种交互式处理程序，但可以提供任意精度算法。它从给定的任何文件中提取输入，然后读取标准输入。

选项:

bc 可识别下列命令行选项:

-c 仅编译。 **bc** 实际上是 **dc** 的预处理器，后者可由 **bc** 自动调用（请参阅 *dc*(1) ）。指定 **-c** 选项可以禁止调用 *dc* ，并且可以将 *dc* 输入发送到标准输出中。

-l 可以预定义任意精度的数学库。其副作用是，需要设置进位制因子。

程序语法:

L 位于范围 **a** 到 **z** 中的单个字母;

E 表达式;

S 语句;

R 关系表达式。

注释:

注释用 **/*** 和 ***/** 括起。

名称:

名称包括:

simple variables: L

array elements: L [E]

单词 **ibase**、 **obase**、和 **scale**

stacks: L

其他操作数

其他操作数包括:

带有可选符号和小数点的任意长度的数字

(E)

sqrt (E)

length (E) 有效的十进制位数

scale (E) 小数点右侧的数位

L (E , ... , E)

用引号 (") 括起的 ASCII 字符的字符串。

算术运算符:

算术运算符生成一个 E 作为结果并包括下列字符:

+ - * / % ^ (**%** 是余数（不是模数，请参阅下文）； ^ 是幂）。

++ -- (前缀和附加项；适用于名称)

= += -= *= /= %= ^=

关系运算符

当作为 **E** *op* **E** 时，关系运算符将生成一个 R:

== <= >= != < >

语句

E

{ S ; ... ; S }

if (R) S

while (R) S

for (E ; R ; E) S

null statement

break

quit

函数定义:

```
define L ( L ,..., L ) {
        auto L, ... , L
        S; ... S
        return ( E )
}
```

–l 数学库中的函数:

-l 数学库中的函数包括:

s(x)	正弦
c(x)	余弦
e(x)	指数
l(x)	对数
a(x)	反正切
j(n,x)	贝塞尔函数

所有函数的参数都由值来传递。依据三角法的角度可以用弧度来表示，其中 2 pi r（弧度）= 360 度。

如果主运算符没有赋值，则将输出语句（它是一个表达式）的值。没有为字符串定义任何运算符，但是，如果字符串出现在应输出表达式结果的环境中，则将输出该字符串。分号或换行符都可以分隔语句。对于采用 *dc*(1) 方式的算术运算，对 进位制的赋值会影响要保留的数位。对 **ibase** 或 **obase** 的赋值将分别设置输入和输出的数值基数，这也由 *dc*(1) 所定义。

可以同时将同一字母用作数组、函数和简单变量。所有的变量对于程序来说都是全局的。在函数调用过程中，"自动"变量将下移。当将数组用作函数参数，或将数组定义为自动变量时，数组名称后面必须跟有空的方括号。

% 运算符根据当前进位制生成余数，而不是整数模数。这样，当进位制为 1 时，**7 % 3** 是 .1（十分之一），而不是 1。这是因为（当进位制为 1 时），**7 / 3** 等于 2.3，.1 是余数。

举例

定义一个函数，以计算指数函数的近似值：

```
scale = 20
define e(x){
        auto a, b, c, i, s
        a = 1
        b = 1
        s = 1
        for(i=1; 1==1; i++){
                a = a*x
                b = b*i
                c = a/b
                if(c == 0) return(s)
                s = s+c
        }
}
```

输出前十个整数的指数函数的近似值。

```
for(i=1; i<=10; i++) e(i)
```

警告

当前没有 **&&** (AND) 或 **||** (OR) 的比较。

for 语句必须具有全部三个表达式。

quit 在读取而不是执行时被解释。

bc 的分析程序在面对输入错误时是不可靠的。一些简单的表达式（例如 2+2）可帮助该分析程序恢复到原始状态中。

赋值运算符： **=+ =- =* =/ =%** 和 **=^** 都已废弃。如果出现这些运算符的任一实例，都会引起语法错误，但 **=-** 除外，因为它可以被解释为后跟一个一元减号的 **=** 。

整个数组和函数都不能作为函数参数来传递。

文件

/usr/bin/dc	桌面计算器可执行程序
/usr/lib/lib.b	数学库

另请参阅

bs(1)、 dc(1)。

《Number Processing Users Guide》中的 *bc* 教程

符合的标准

bc： XPG4、POSIX.2

名称

bdiff - 用于大型文件的 diff 命令

概要

bdiff *file1 file2* [*n*] [**-s**]

说明

bdiff 比较两个文件，并与 **diff** （请参阅 *diff*(1) ）命令生成相同的输出，同时指定使这两个文件相同而必须进行的更改。 **bdiff** 用于处理对于 **diff** 来说太大的文件，不过它可用于比较任意长度的文件。

bdiff 按以下方式处理文件：

- 忽略两个文件开头的相同行。
- 将每一文件的剩余部分划分为多个含有 *n* 行的段，然后对相应段执行 **diff** 。 *n* 的缺省值为 3500。

命令行参数

bdiff 可识别下列命令行参数：

file1
file2 **bdiff** 将比较的两个文件的名称。如果 *file1* 或 *file2* （但不同时）为 **-** ，则使用标准输入。

n 如果存在一个数值作为第三个参数，则文件将被分为多个含有 *n* 行的段，然后再由 **diff** 处理。 *n* 的缺省值为 3500。当 3500 行的段太大以致 **diff** 无法处理时，可以使用该选项。

-s 静默选项禁止 **bdiff** 输出诊断消息（但允许 **diff** 输出可能的错误消息）。如果同时指定了 *n* 和 **-s** 参数，则在命令行中 *n* 参数必须位于 **-s** 选项之前，否则将不能正确识别该参数。

外部语言环境影响

环境变量

LC_MESSAGES 用于确定显示消息的语言。

如果在环境中未指定 **LC_MESSAGES** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 。

如果任一国际化变量中包含无效设置，则 **bdiff** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

both files standard input (bd2)

为两个文件都指定了标准输入。仅有一个文件可以指定为标准输入。

non-numeric limit (bd4)

为 *n* （第三个）参数指定了非数字值。

举例

查找以下两个大型文件之间的差异： **file1** 和 **file2** ，并将结果放置在名为 **diffs_1.2** 的新文件中。

bdiff file1 file2 >diffs_1.2

执行相同操作，但是将文件长度限制为 1400 行；禁止输出错误消息：

bdiff file1 file2 1400 -s >diffs_1.2

警告

bdiff 与 **diff** 生成相同的输出，并进行必需的行数调整，以便使输出形式类似于由 **diff** 处理的。但是， **bdiff** 可能找到，也可能找不到一组完全最小化的文件差异，这取决于文件拆分的位置。

文件

/tmp/bd??????

另请参阅

diff(1)。

名称

bs - 中等大小程序的编译程序/解释程序

概要

bs [*file* [*args*]]

说明

bs 是 BASIC 和 SNOBOL4 的远程子代，添加了一些 C 语言。设计 **bs** 是用于完成某些编程任务，在这些任务中程序开发时间与得到的执行速度一样重要。它最大限度地减少了数据声明和文件（或进程）处理的规定。一次一行的调试方法、 **trace** 和 **dump** 语句以及有用的运行时错误消息，所有这一切都简化了程序测试。此外，可以调试不完整的程序；在编写 外部函数之前可以测试 内部函数，反之亦然。

如果在命令行上指定了 *file* ，则它用于从键盘进行任何输入之前的输入。缺省情况下，编译从 *file* 读取的语句以便稍后执行。同样，通常立即执行从键盘输入的语句（请参阅下面的 **compile** 和 **execute** ）。除非最终操作为赋值，否则将打印最接近的表达式语句的结果。

bs 程序由输入行组成。如果某行上的最后一个字符是 **** ，则该行将延续到下一行。 **bs** 接受以下格式的行：

> *statement*
> *label statement*

标签是后跟一个冒号的 *name* （见下文）。标签和变量可以具有相同的名称。

bs 语句是一个表达式或后跟零个或多个表达式的关键字。一些关键字 （**clear**、 **compile**、 **!**、 **execute**、 **include**、 **ibase**、 **obase** 和 **run**）始终在被编译时执行。

语句语法

expression 为其副作用执行该表达式（值、赋值或函数调用）。表达式的详细信息遵循下面的语句类型说明。

break **break** 从最内层 **for/while** 循环退出。

clear 清除符号表和已编译的语句。立即执行 **clear** 。

compile [*expression*]
编译后续语句（覆盖立即执行缺省值）。计算可选表达式，并将其用作文件名以供进一步输入。**clear** 与此后一种情况关联。立即执行 **compile** 。

continue **continue** 转移到当前 **for/while** 循环的循环延续。

dump [*name*] 打印每个非局部变量的名称和当前值。也可以仅报告指定的变量。在出现错误或中断后，将显示最后一条语句的编号。在函数中出现错误或 **stop** 后，将显示用户函数跟踪。

edit 如果由 **EDITOR** 环境变量选择的编辑器存在，则对它发出调用；如果未定义 **EDITOR** 或它为空，则对 *ed*(1) 发出调用。如果命令行上存在 *file* 参数，则将 *file* 作为要编辑的文件传递到编辑器（否则，不使用文件名）。在退出编辑器时，执行 **compile** 语句（以及关联的 **clear** ），将该文件名作为其参数提供。

exit [*expression*] 返回到系统级别。将表达式作为进程状态返回。

execute 更改为立即执行模式（中断具有类似的效果）。此语句不会导致执行存储的语句（请参阅下面的 **run** ）。

for *name* **=** *expression expression statement*

for *name* **=** *expression expression*

...

next

for *expression* **,** *expression* **,** *expression statement*

for *expression* **,** *expression* **,** *expression*

...

next 在指定变量的控制下， **for** 语句重复执行语句（第一种形式）或语句组（第二种形式）。变量采用第一个表达式的值，然后在每个循环中加 1，但不超过第二个表达式的值。第三和第四种形式需要三个用逗号分隔的表达式。其中第一个表达式是初始化，第二个是测试（如果为 true，则继续），第三个是循环延续操作（通常为增量）。

fun *f*([*a*, ...]) [v, ...]

...

nuf **fun** 定义用户编写函数的函数名称、参数和局部变量。最多允许十个参数和局部变量。这样的名称不能是数组，也不能与 I/O 关联。不能嵌套函数定义。允许调用未定义的函数；请参阅下面的函数调用。

freturn 指示用户编写的函数失败的方式。请参阅下面的问号运算符 **(?)** 。如果问号不存在，则 **freturn** 仅返回零。当问号 处于活动状态时， **freturn** 将转移到该表达式（可能跳过中间的函数返回）。

goto *name* 控制权将被传递到具有匹配标签的内部存储语句。

ibase *n* **ibase** 将输入基数设置为 *n*。仅支持 *n* 的以下值：常数 **8**、 **10** （缺省值）和 **16**。十六进制值 10-15 输入为 **a-f** 。前导数字是必需的（即，必须将 **f0a** 输入为 **0f0a** ）。将立即执行 **ibase** （以及下面讨论的 **obase** ）。

if *expression statement*

if *expression*

...

[**else**...]

fi 如果表达式的计算结果不为零，则执行语句（第一种形式）或语句组（第二种形式）。字符串 **0** 和 **""** （空值）计算为零。在第二种形式中，可选的 **else** 提供在不执行第一个语句组时要执行的第二个语句组。在具有 **else** 的同一行上允许的唯一语句是 **if** ；只有其他 **fi** 可以在具有 **fi** 的同一行上。支持将 **else** 和 **if** 连接为 **elif** 。。只需一个 **fi** 即可结束 **if** ... **elif** ... [**else** ...] 序列。

include *expression*
: *expression* 的计算结果必须是一个文件名。该文件必须包含 **bs** 源语句。这样的语句将成为正编译的程序的一部分。不能嵌套 **include** 语句。

obase *n* — **obase** 将输出基数设置为 *n* （请参阅上面的 **ibase** ）。

onintr *label*
onintr — **onintr** 提供了中断的程序控制。在第一种形式中，控制权传递到给定的标签，就像在执行 **onintr** 时已执行 **goto** 。。在每个中断后清除语句的效果。在第二种形式中，中断可导致 **bs** 终止。

return [*expression*]
: 计算表达式，并将结果作为函数调用的值传递回来。如果未指定表达式，则返回零。

run — 重置随机编号生成器。控制权将被传递到第一条内部语句。如果 **run** 语句包含在文件中，则它应该是最后一条语句。

stop — 停止内部语句的执行。 **bs** 还原为即时模式。

trace [*expression*]
: **trace** 语句控制函数跟踪。如果表达式为空（或其计算结果为零），则将关闭跟踪。否则，将打印用户函数调用（或返回）的记录。每个 **return** 都会将 **trace** *expression* 的值减 1。

while *expression statement*
while *expression*
...
next — **while** 与 **for** 类似，不同之处是仅提供循环延续的条件表达式。

! *Shell command* — 立即退出到 Shell。

... — 忽略此语句（将其视为注释）。

表达式语法

name — 名称用于指定变量。名称由一个字母（大写或小写）以及（可选）后跟的字母和数字组成。只有名称中的前六个字符才是有意义的。除了在 *fun* 语句中声明的名称外，所有名称对于程序都是全局性的。名称可以采用数字（双精度浮点）值、字符串值，也可以与输入/输出关联（请参阅下面的内置函数 **open()** ）。

name ([expression [, expression] ...])
: 可以通过后跟圆括号中用逗号分隔的参数的名称调用函数。除了内置函数（在下面列出）外，必须使用 *fun* 语句定义名称。函数的参数是按值传递的。如果未定义函数，则打印调用该函数的调用历史记录，并请求返回值（作为表达式）。将该表达式的结果作为未定义函数的结果。这样，就可以调试尚未定义所有函数的程序了。从当前输入文件读取值。

name [expression [, expression] ...]
: 此语法用于引用数组或表（请参阅下面的内置 *table* 函数）。对于数组，将每个表达式截断为整数，并用作名称的说明符。得到的数组引用在语法上与名称相同； **a[1,2]** 与 **a[1][2]** 相同。被截

断表达式的值被限制在 0 到 32767 之间。

number　　数值用于表示常量值。数值是以 Fortran 样式书写的，它包含数字、可选小数点，还可能包含由 **e** 及后跟的可能有符号指数组成的比例因子。

string　　字符串由 **"** 字符定界。 **** 转义字符允许双引号 (**\"**)、换行符 (**\n**) 、回车符 (**\r**) 、退格符 (\b) 和制表符 (**\t**) 等字符出现在字符串中。否则， **** 表示它自己。

(expression)　　圆括号用于更改计算的正常顺序。

(expression , expression [, expression ...]) [expression]

括起来的表达式用作下标，以便从括起来的列表中选择逗号分隔的表达式。列表元素从左侧编号，且从零开始编号。

表达式：

(False, True)[a == b]

如果比较结果为 true，则具有值 **True** 。

? expression　　问号运算符测试表达式是否成功而不是测试其值。此时，测试文件结束标志（请参阅下面的 编程提示一节中的示例）、 **eval** 内置函数的结果和检查用户编写函数的返回值（请参阅 **freturn** ）是很有用的。询问“陷阱”（文件结束标志等）会导致立即转移到最近的询问，可能会跳过赋值语句或干涉函数级别。

- expression　　结果是对表达式求反。

++ name　　将变量（或数组引用）的值加 1。结果为新值。

-- name　　将变量的值减 1。结果为新值。

!expression　　表达式的逻辑“非”。请谨慎使用 Shell 转义命令。

expression　　*operator* 表达式两个参数的公共函数由表示该函数的运算符分隔的两个参数进行缩写。除了赋值、连接和关系运算符外，在应用函数之前，这两个操作数都将被转换为数字格式。

二元运算符

按优先级递增的顺序。

=　　**=** 是赋值运算符。左操作数必须是一个名称或数组元素。结果是右操作数。赋值从右向左绑定，所有其他运算符都是从左向右绑定。

_　　_ （下划线）是连接运算符。

& |　　**&** （逻辑“与”）的任一参数为零时，它的结果为零。如果它的这两个参数都不为零，则它的结果为 1；如果其参数为零，则 **|** （逻辑“或”）的结果为零。如果其任一参数都不为零，则它的结果为 1。这两个运算符都将空字符串视为零。

< <= > >= == !=
如果其参数在指定的关系中，则关系运算符 （**<**：小于， **<=** ：小于或等于， **>** ：大于， **>=**：大于或等于， **==**：等于， **!=**：不等于）返回 1；否则返回零。处于相同级别的关系运算符扩展如下： **a>b>c** 等同于 *a>b & b>c* 。如果这两个操作数都是字符串，则进行字符串比较。

+ - 加和减。

*** / %** 乘、除和余数。

^ 取幂。

内置函数

处理参数

arg(*i*) 是当前级别函数调用上第 *i* 个实际参数的值。在零级上， *arg* 返回第 *i* 个命令行参数 （*arg*(0) 返回 **bs** ） 。

narg() 返回传递的参数数目。在零级上，返回命令参数计数。

数学

abs(*x*) 是 *x* 的绝对值。

atan(*x*) 是 *x* 的反正切。其值介于 -Pi/2 和 Pi/2 之间。

ceil(*x*) 返回不小于 *x* 的最小整数。

cos(*x*) 是 *x* （弧度）的余弦。

exp(*x*) 是 *x* 的指数函数。

floor(*x*) 返回不大于 *x* 的最大整数。

log(*x*) 是 *x* 的自然对数。

rand() 是在 0 和 1 之间均匀分布的随机数。

sin(*x*) 是 *x* （弧度）的正弦。

sqrt(*x*) 在 *x* 的平方根。

字符串运算

size(*s*) 返回 *s* 的大小（以字节为单位的长度）。

format(*f*，*a*) 返回 *a* 的格式化值。假定 *f* 为 *printf*(3S) 样式的格式规范。只有 **%...f**、 **%...e** 和 **%...s** 类型是安全的。由于在编写 **format** 调用的代码时并不是总有可能知道 **a** 是一个数字还是字符串，因此应该考虑将 **a** 强制为 **f** 所需的类型，方法是添加零（对于 **e** 或 **f** 格式）或连接 (_) 空字符串（对于 **s** 格式）。

index(*x*，*y*) 返回与 *y* 中的任一字符匹配的 *x* 中第一个位置的编号。如果不匹配，则产生零。

trans(*s*, *f*, *t*) 将源 *s* 中的字符从 *f* 中的匹配字符转换为 *t* 中相同位置中的字符。将不出现在 *f* 中的源字符复制到结果中。如果字符串 *f* 比 *t*2005-7-22*9:37* 长，则在 *f* 的超出部分中匹配的源字符不出现在结果中。

substr(*s*, *start*, *width*)

返回由 *start* 位置和 *width* 定义的 *s* 的子字符串。

match(*string*, *pattern*)

mstring(*n*) *pattern* 是按照基本正则表达式定义的正则表达式（请参阅 *regexp*(5)）。 **mstring** 返回在对 *match* 最新调用时在模式符号对 **\\(** 和 **\\)** 之间出现的主题的第 *n* (1 <= *n* <= 10) 个子字符串。要获得成功，模式必须与字符串的开头匹配（好像所有模式都以 ^ 开头）。函数返回匹配的字符数。例如：

```
match("a123ab123", ".*\([a-z]\)") == 6
mstring(1) == "b"
```

文件处理

open(*name*, *file*, *function*)

close(*name*) *name* 参数必须是一个 **bs** 变量名称（作为字符串传递）。对于 **open**， *file* 参数可以是：

1. 0（零）、1 或 2，分别表示标准输入、输出或错误输出；
2. 表示文件名的字符串；或
3. 以 **!** 开头的字符串，表示要执行（通过 **sh -c** ）的命令。 *function* 参数必须是 **r** （读取）、 **w** （写入）、 **W** （写入但不换行）或 **a** （追加）。在 **close** 后， *name* 将还原为普通变量。如果 *name* 是一个管道，则在关闭完成之前执行 **wait()** （请参阅 *wait*(2)）。 **bs exit** 命令不进行这样的等待。初始关联是：

```
open("get", 0, "r")
open("put", 1, "w")
open("puterr", 2, "w")
```

在下一节中提供了示例。

access(*s*, *m*) 执行 **access()** （请参阅 *access*(2)）。

ftype(*s*) 返回单个字符文件类型指示： **f** 表示常规文件， **p** 表示 FIFO （即命名管道）， **d** 表示目录， **b** 表示块设备专用， **c** 表示字符设备专用。

表

table(*name*, *size*)

bs 中的表是关联访问的单维数组。“下标”（称为键）是字符串（数字会被转换）。 *name* 参数必须是一个 **bs** 变量名称（作为字符串传递）。 *size* 参数设置要分配的最小元素数。 **bs** 打印错误消息并在表溢出时停止。 *table* 的结果是 *name* 。

item(*name*, *i*)

key() **item** 函数顺序访问表元素（在正常使用情况下，键值不会按顺序排列）。其中 **item** 函数访问值， **key** 函数访问以前 **item** 调用的“下标”。如果以前 **item** 调用没有有效的下标，则它将失败（或者在缺少 **interrogate** 运算符时返回空值）。 *name* 参数不应用引号括起来。由于未定义精确的表大小，因此应该使用问号运算符检测表结束标志；例如：

```
table("t", 100)
    ...
# If word contains "party", the following expression adds one
# to the count of that word:
++t[word]
    ...
# To print out the the key/value pairs:
for i = 0, ?(s = item(t, i)), ++i if key() put = key()_":"_s
```

在未使用问号运算符的情况下，如果表中没有另外的元素，则 **item** 的结果为空。但是，空值是合法的“下标”。

iskey(*name***,** *word***)**
iskey 测试键 *word* 是否存在于表 *name* 中，并返回 1（如果为 true）或 0（如果为 false）。

其他未尽事项

eval(*s***)** 字符串参数计算为 **bs** 表达式。对于将数字字符串转换为数字内部格式，该函数是很方便的。**eval** 也可以用作间接引用的粗略形式，例如：

```
name = "xyz" eval("++"_ name)
```

将变量 *xyz* 加 1。此外，前面有问号运算符的 **eval** 允许用户控制 **bs** 错误条件。例如：

```
?eval("open(\"X\", \"XXX\", \"r\")")
```

如果没有名为 **XXX** 的文件，则返回零值（而不是暂停用户的程序）。下例执行 **goto** 直到标签 **L** （如果它存在）：

```
label="L"
if !(?eval("goto "_ label)) puterr = "no label"
```

plot(*request***,** *args***)**
如果 **tplot** 命令可用，则 **plot** 函数在 **tplot** 识别的设备上产生输出。 *request* 如下所示：

Call	**Function**
plot(0, *term***)**	导致更多带有 **-T***term* 参数的 *plot* 输出传输到 *tplot* 中。 *term* 的长度最大为 40 个字符。
plot(1)	“清除”绘图仪。

plot(2, *string***)**	用 *string* 标记当前点。
plot(3, *x1, y1, x2, y2***)**	绘制 (*x1*,*y1*) 和 (*x2*,*y2*) 之间的线。
plot(4, *x, y, r***)**	绘制一个圆，其圆心是 (*x*,*y*)，半径是 *r* 。
plot(5, *x1, y1, x2, y2, x3, y3***)**	绘制一个圆弧（按逆时针方向），圆心为 (*x1*,*y1*)，端点为 (*x2*,*y2*) 和 (*x3*,*y3*)。
plot(6)	未实现。
plot(7, *x, y***)**	创建当前点 (*x*,*y*)。
plot(8, *x, y***)**	绘制从当前点到 (*x*,*y*) 的线。
plot(9, *x***,** *y***)**	在 (*x*,*y*) 处绘制一个点。
plot(10, *string***)**	将行模式设置为 *string* 。
plot(11, *x1, y1, x2, y2***)**	使 (*x1*,*y1*) 成为绘图区的左下角，并使 (*x2*,*y2*) 成为绘图区的右上角。
plot(12, *x1, y1, x2, y2***)**	将后续的 x (y) 坐标乘以 *x1* (*y1*) ，然后在绘制它们之前将其与 *x2* (*y2*) 相加。初始比例是 **plot(12, 1.0, 1.0, 0.0, 0.0)** 。

某些请求不适用于所有绘图仪。除 0 和 12 之外的所有请求都是通过将字符传输到 *tplot* 实现的。

从键盘执行的每条语句重新调用 **tplot** ，如果没有在单个操作中绘制完整的图片，则会使结果变得不可预测。因此，应该在函数或完整程序中完成绘制操作，以便可以在单个数据流中将所有输出定向到 **tplot** 。

last() 在即时模式下， **last** 返回最近计算的值。

外部语言环境影响

环境变量

LC_COLLATE 确定在计算正则表达式时使用的排序序列。

LC_CTYPE 确定与正则表达式中字符类表达式匹配的字符。

如果未在环境中指定 **LC_COLLATE** 或 **LC_CTYPE**，或者将其设置为空字符串，则会将 **LANG** 的值用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或者将它设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5) ），而不使用 **LANG**。如果任一国际化变量中包含无效设置，则 **bs** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集。

举例

将 **bs** 用作计算器（**$** 是 Shell 提示符）：

```
$ bs
# Distance (inches) light travels in a nanosecond.
186000 * 5280 * 12 / 1e9
11.78496
     ...

# Compound interest (6% for 5 years on $1,000).
int = .06 / 4
bal = 1000
for i = 1 5*4 bal = bal + bal*int
bal - 1000
346.855007
     ...
exit
```

典型 **bs** 程序的概要:

```
# initialize things:
var1 = 1
open("read", "infile", "r")
     ...
# compute:
while ?(str = read)
     ...
next
# clean up:
close("read")
     ...
# last statement executed (exit or stop):
exit
# last input line:
run
```

输入/输出示例:

```
# Copy file oldfile to file newfile.
open("read", "oldfile", "r")
open("write", "newfile", "w")
     ...
while ?(write = read)
```

```
      ...
# close "read" and "write":
close("read")
close("write")

# Pipe between commands.
open("ls", "!ls *", "r")
open("pr", "!pr -2 -h 'List'", "w")
while ?(pr = ls) ...
      ...
# be sure to close (wait for) these:
close("ls")
close("pr")
```

警告

除非 **tplot** 命令在您的系统上可用，否则图形模式 (**plot** ...) 不是特别有用。

bs 无法容忍某些错误。例如，错误键入 **fun** 声明后是很难更正的，因为不执行 **clear** 就无法建立新定义。在这样的情况下，最好的解决方法是从使用 **edit** 命令开始。

另请参阅

ed(1)、 sh(1)、 access(2)、 printf(3S)、 stdio(3S)、 lang(5)、 regexp(5)。

有关数学函数的进一步说明，请参阅 Section (3M)。

pow() 用于取幂 — 请参阅 *exp*(3M) ）；

bs 使用标准的 I/O 包。

名称

cal - 输出日历

概要

cal [[*month*] *year*]

说明

cal 输出指定年份的日历。如果还指定了月份，则仅输出该月的日历。如果年和月都未指定，则输出当前月的日历。 *year* 可介于 1 到 9999 之间。 *month* 是介于 1 和 12 之间的十进制整数。生成的日历是 Gregorian 历（即现在的阳历）。

外部语言环境影响

环境变量

LANG 确定当 **LC_ALL** 和相应的环境变量（以 **LC_** 开头）都未指定语言环境时，语言环境类别将使用的语言环境。如果未设置 **LANG** 或将其设置为空字符串，则使用缺省值"C"（请参阅 *lang*(5) ）。

LC_CTYPE 将用于解释文本数据字节序列的语言环境定义为字符（例如，参数文件和输入文件中的单字节和多字节字符）。

LC_TIME 确定日历的格式和内容。

TZ 确定用于计算当前月份的值的时区。

如果任一国际化变量包含无效设置，则 **cal** 将认为所有国际化变量都设置为"C"。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

命令：

cal 9 1850

在屏幕上输出 1850 年 9 月的日历，如下所示：

September 1850

S	M	Tu	W	Th	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

但对于 XPG4，其输入结果如下所示：

Sep 1850

Sun	Mon	Tue	Wed	Thu	Fri	Sat

1	**2**	**3**	**4**	**5**	**6**	**7**
8	**9**	**10**	**11**	**12**	**13**	**14**
15	**16**	**17**	**18**	**19**	**20**	**21**
22	**23**	**24**	**25**	**26**	**27**	**28**
29	**30**					

警告

年份始终被认为从 1 月开始，即使从历史角度来说并不现实。

请注意， **cal 83** 指的是基督纪元的 83 年，而不是 20 世纪的 83 年。

符合的标准

cal：SVID2、SVID3、XPG2、XPG3、XPG4

名称

calendar - 提醒服务

概要

calendar [-]

说明

calendar 查询当前目录中的文件 **calendar** ，并输出在行中任意位置包含今天或明天日期的行。在周末，“明天”将顺延到星期一。

如果存在 **-** 命令行参数，则 **calendar** 会在每个用户的主目录中搜索文件 **calendar** ，并将任何正面结果通过 **mail** （请参阅 *mail*(1) ）发送给用户。这一工作通常是每天清晨在 **cron** 的控制下进行的（请参阅 *cron*(1M) ）。当被 **cron** 调用时， **calendar** 将读取 **calendar** 文件中的第一行以确定用户的环境。

诸如拼写和日期格式之类的语言相关信息（如下所述）由 **calendar** 文件中用户指定的 **LANG** 语句确定。此语句的格式应该为 **LANG=***language* ，其中 *language* 是一种有效的语言名称（请参阅 *lang*(5) ）。如果 **calendar** 文件中不存在此行，则会根据“外部语言环境影响”的“环境变量”部分中的说明指定语言。

calendar 与以下两个字段有关：月份字段和日期字段。月份字段可以按以下三种不同格式表示：表示月份名称的字符串（全拼或缩写）、数字月份或星号（表示任何月份）。如果用表示月份名称的字符串来表示月份，则首字符可以大写，也可以小写；而其他字符必须小写。月份名称的拼写形式应该与调用 **nl_langinfo()** 时返回的字符串匹配（请参阅 *nl_langinfo*(3C) ）。日期字段是一月中某日的数字值。

月份**-**日期格式

如果月份字段是一个字符串，则其后可以跟零个或多个空白字符。如果月份字段是一个数字，则其后必须跟一个斜线 (**/**) 或连字符 (**-**) 。如果月份字段是一个星号 (*****) ，则必须后跟一个斜线 (**/**) 。日期字段的后面可以紧跟空白或非数字字符。

日期**-**月份格式

将日期字段表示为一个数字。日期字段后面的内容由月份的格式决定。如果月份字段是一个字符串，则日期字段必须后跟零个或一个点 (**.**) ，然后接零个或多个空白字符。如果月份字段是一个数字，则日期字段必须后跟一个斜线 (**/**) 或连字符 (**-**) 。如果月份字段是一个星号，则日期字段必须后跟一个斜线 (**/**) 。

外部语言环境影响

环境变量

如果在 **calendar** 文件中未指定 **LANG** 语句，则 **LC_TIME** 将用于确定日期和时间字符串的格式和内容。

LANG 用于确定显示消息的语言。

如果未在环境中指定 **LC_TIME** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **calendar** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

下面的 **calendar** 文件说明了 *calendar* 可识别的几种格式：

LANG=en_US.roman8
Friday, May 29th: group coffee meeting
meeting with Boss on June 3.
3/30/87 - quarter end review
4-26 Management council meeting at 1:00 pm
It is first of the month (*/1); status report due.

在下面的 **calendar** 文件中，日期是按照欧洲英语用法表示的：

LANG=en_GB.roman8
On 20 Jan. code review
Jim's birthday is on the 3. February
30/3/87 - quarter end review
26-4 Management council meeting at 1:00 pm
It is first of the month (1/*); status report due.

警告

要获得提醒服务，您的日历必须是公共信息，或者您必须从个人的 **crontab** 文件运行 **calendar** ，独立于在系统级内运行的任何 **calendar -** 。请注意，如果您自己运行 **calendar** ，则日历文件无需位于您的主目录录中。

calendar 的"明天"顺延方法不适用于假日。

X/Open 标准中可能会取消此命令。使用此命令的应用程序可能无法移植到其他供应商的平台上。

作者

calendar 由 AT&T 和 HP 开发。

文件

calendar
/tmp/cal*
/usr/lbin/calprog 确定今天和明天的日期
/usr/bin/crontab
/etc/passwd

另请参阅

cron(1M)、 nl_langinfo(3C)、 mail(1)、 environ(5)。

符合的标准

calendar： SVID2、SVID3、XPG2、XPG3

名称

cat - 连接、复制和打印文件

概要

cat [**-benrstuv**] *file* ...

说明

cat 按顺序读取每个 *file* ，并将其写入标准输出。因此：

cat *file*

在缺省标准输出设备上打印 *file* ；

cat *file1 file2* **>** *file3*

连接 *file1* 和 *file2* ，然后将结果放入 *file3* 。

如果 **-** 显示为 *file* 参数， **cat** 将使用标准输入。要组合标准输入和其他文件，请使用 **-** 和 *file* 参数的组合。

选项

cat 可识别下列选项：

-b 在指定 **-n** 选项时省略空白行的行号。如果指定该选项，则将自动选择 **-n** 选项。

-e 在每行的末尾打印一个 **$** 字符（在换行符之前）。如果指定该选项，则将自动选择 **-v** 选项。

-n 显示前面带有行号的输出行（从 1 开始按顺序编号）。

-r 将多个连续的空行替换为一个空行，使得包含字符的行之间决不会有多个空行。

-s 静默选项。 **cat** 禁止有关不存在的文件、等同的输入和输出以及写入错误的错误消息。通常，输入和输出文件不能同名（除非是特殊文件）。

-t 将每个制表符打印为 **^I** ，将换页符打印为 **^L** 。如果指定该选项，则将自动选择 **-v** 选项。

-u 不要缓冲输出（逐个字符地处理）。通常情况下会缓冲输出。

-v 以可见的形式打印非打印字符（制表符、换行符和换页符除外）。控制符以 ^*X* (Ctrl-*X*) 形式打印，DEL 字符（八进制数 0177）打印为 **^?** （请参阅 *ascii*(5) ）。已设置其最重要位的单字节控制符使用 **M-^***x* 形式进行打印，其中 *x* 是由七个低次位指定的字符。其他所有非打印字符均打印为 **M-***x* ，其中 *x* 是由七个低次位指定的字符。该选项受到 **LC_CTYPE** 环境变量及其相应字符集的影响。

外部语言环境影响

环境变量

LANG 为没有设置或设置为 null（空）的国际化变量提供了缺省值。如果 **LANG** 未设置或设置为 null（空），则会使用缺省值“C”（请参阅 *lang*(5) ）。如果任一国际化变量中包含无效设置，则 **cat** 的行为类似于所有国际化变量都设为“C”。请参阅 *environ*(5) 。

LC_ALL 如果设为非空字符串值，则会覆盖所有其他国际化变量的值。

LC_CTYPE 确定文本作为单字节和（或）多字节字符的解释、字符作为可打印字符的分类以及在常规表达式中按字符类表达式匹配的字符。

LC_MESSAGES 确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息以及写入到标准输出的信息消息的格式和内容。

NLSPATH 确定消息目录的位置，以便处理 **LC_MESSAGES** 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

退出值包括：

0 成功完成。

>0 发生了错误。

举例

要创建长度为零的文件，可使用以下其中任一命令：

cat */dev/null* **>** *file*
cp */dev/null file*
touch *file*

以下命令为制表符在 *file1* 中的所有实例打印 **^I**

cat -t *file1*

要禁止有关不存在的文件的错误消息，请使用：

cat -s *file1 file2 file3* **>** *file*

如果不存在 *file2* ，以上命令将连接 *file1* 和 *file3* ，而不报告有关 *file2* 的错误。该结果与不使用 **-s** 选项时相同，例外的是 **cat** 会显示错误消息。

要查看 *file2* 中的非打印字符，请使用：

cat -v *file2*

警告

如下命令格式

cat *file1 file2* **>** *file1*

将在连接开始之前覆盖 *file1* 中的数据，从而损坏该文件。因此，使用 Shell 特殊字符时应谨慎小心。

另请参阅

cp(1)、 more(1)、 pg(1)、 pr(1)、 rmnl(1)、 ssp(1)。

符合的标准

cat：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

cc_bundled： cc - 捆绑的 C 编译程序

概要

备注

对于基于 Itanium® 的系统，请参阅 *cc_bundled_ia*(1) 。

对于 PA-RISC 系统，请参阅 *cc_bundled_pa*(1) 。

使用 **uname** 命令确定您的系统类型。在基于 Itanium 的系统上， **uname -m** 返回 **ia64** 。所有其他值表示 PA-RISC 系统。

另请参阅

cc_bundled_ia(1)、 cc_bundled_pa(1)、 uname(1)。

名称

cc_bundled_ia： cc - 捆绑的 C 编译程序

概要

cc [*options*] *files*

说明

本联机帮助页描述捆绑的 C 编译程序。 **cc** 调用 HP-UX 捆绑的 **C** 编译程序。 C 源代码直接编译为对象代码。

命令使用 **ctcom** （基于 Itanium® 的系统）或 **ccom** （PA-RISC，精度体系结构系统）编译程序进行预处理、语法和类型检查，以及代码生成。

cc 接受几种类型的参数作为 *files* ：

.c 后缀　将名称以 **.c** 结尾的参数识别为 C 源文件。每个 **.c** 文件都被编译，所得到的对象文件将置于一个具有相应基名的文件中，但其名称后缀是 **.o** 而不是 **.c** 。但是，如果编译了一个 C 源文件并一步完成所有链接，则 **.o** 文件将被删除。

.s 后缀　将名称以 **.s** 结尾的参数识别为汇编源文件并执行汇编，从而为每个 **.s** 文件生成一个 **.o** 文件。

.i 后缀　将名称以 **.i** 结尾的参数视为预处理的源文件。（请参阅下面的 **-P** 选项）每个对象文件会置于具有相应名称的文件中，但其名称后缀为 **.o** 而不是 **.i** 。

-l*x* Form　**-l***x* 格式的参数会使链接程序搜索库 **lib***x***.so** （基于 Itanium 的系统）或 **lib***x***.sl** ，（PA-RISC 系统）或 **lib***x***.a** 以试图解析当前未解析的外部引用。由于在遇到库名时对库进行搜索，因此 **-l** 的放置位置很重要。如果文件包含一个未解析的外部引用，则在命令行上必须将包含定义的库放置在该文件 之后。有关其他信息，请参阅 *ld*(1) 。

-l:lib*x***.***suffix* Form

-l:lib*x***.***suffix* 格式的参数会使链接程序搜索库 **lib***x***.so** （基于 Itanium 的系统）或 **lib***x***.sl** （PA-RISC 系统）或 **lib***x***.a** （取决于 *suffix* ），以试图解析当前未解析的外部引用。它与 **-l** 选项类似，但例外的是 **-Wl,-a** 选项的当前状态不重要。

Other Suffixes　所有其他参数（如那些名称以 **.o** 、 **.a** 或 **.so** 结尾的参数）将被视为浮动对象文件，并传递到 **ld** （请参阅 *ld*(1) ）以包含在链接操作中。

通过 **CCOPTS** 环境变量和命令行操作，可以将参数和选项传递到编译程序。编译程序读取 **CCOPTS** 的值并将这些选项分为两组；一组选项位于竖线 (|) 之前并用空白字符分隔，另一组选项位于竖线之后。第一组选项放在 **cc** 的所有命令行参数之前；第二组选项放在 **cc** 的命令行参数之后。如果竖线不存在，则所有选项都放在命令行参数之前。例如（在 *sh*(1) 表示法中），

```
CCOPTS=-v | -lm
export CCOPTS
cc -p prog.c
```

等同于

cc -v -p prog.c -lm

设置后， **TMPDIR** 环境变量会指定编译程序要使用的、用于存放临时文件的目录，从而覆盖缺省目录 **/var/tmp**。

选项

注释：有关任何选项的详细信息，请参阅《HP C Online Help》。要调用联机帮助，请使用 HTML 浏览器打开 URL **file:/opt/ansic/html/$LANG/guide/index.htm** （基于 Itanium 的系统）或 **file:/opt/ansic/html/guide/$LANG/index.htm** （PA-RISC 系统）。

cc 可识别下列选项：

-b 使链接程序 **ld** 创建一个共享库，而不是普通的可执行文件。有关详细信息，请参阅 *ld*(1) 、《 HP-UX Linker and Libraries Online User's Guide 》和《 Programming on HP-UX 》手册。

-c 禁止编译的链接编辑阶段，并强制为每个 **.c** 或 **.i** 文件生成一个对象 (**.o**) 文件（即使只编译了一个程序）。从 C 程序生成的对象文件必须先链接，然后再执行。

-D*name*=*def*

-D*name* 将 *name* 定义为预处理器传递（属于 **ctcom** ），与使用“#define”相同。

-E 预处理命名的 C 文件，并将结果发送到标准输出。

-.*suffix* 如果不使用 **-E** 选项的标准输出，则将每个 **.c** 文件的输出放入具有相应的 *.suffix* 的文件中。

-I*dir* 更改预处理器传递使用的算法，以查找同样要在 *dir* 中搜索的包含文件。请参阅《 *HP C Online Help* 》。

-I- 包含文件的目录不再用作查找用双引号 () 括起的文件的起点。它们从第一个 **-I** 开始查找。用 **<>** 括起的文件将通过跟在 **-I-** 后面的路径进行查找。

-l*x* 请参阅 说明部分中的 **-l***x* 和 **-l:lib***x***.***suffix* 的说明（第四条和第五条）。

-L *dir* 更改链接程序用于搜索 **lib***x***.so** （基于 Itanium 的系统）或 **lib***x***.sl** （PA-RISC 系统）或 **lib***x***.a** 的算法。 **-L** 选项会使 *ld* 先在 *dir* 中搜索，然后在缺省位置中搜索。仅当在命令行上此选项位于 **-l** 选项之前时，此选项才有效。有关详细信息，请参阅 *ld*(1) 。

-o*outfile* 指定 *outfile* 作为链接程序输出文件的名称。缺省名称为 **a.out** 。当与 **-b** 一起使用时，指定共享库的名称。当与 **-c** 一起使用时，指定对象文件的名称。

-P 预处理命名的 C 文件，并将结果置于后缀为 **.i** 的相应文件中。

-s 导致去除链接程序输出结果的符号表信息。有关详细信息，请参阅 *strip*(1) 。使用该选项将禁止对生成的程序使用符号调试程序。有关详细信息，另请参阅 *ld*(1) 。

如果 **-s** 是与其他任何选项一同指定的，则无论指定这些选项的顺序如何，都会忽略 **-s** 选项。

-S 编译命名的 C 文件，并将汇编语言输出结果置于后缀为 **.s** 的相应文件中。

-t*x*,*name* 使用 *name* 替换或插入子进程 *x* ，其中 *x* 是一组指示子进程的标识符中的一个或多个标识符。该选项的工作模式有两种：1) 如果 *x* 是单个标识符，则 *name* 表示新子进程的完整路径名；2) 如果 *x* 是一组标识符，则 *name* 表示一个前缀，该前缀将连接标准后缀，以构建新的子进程的完整路径名。

x 可以使用以下一个或多个值：

a 汇编程序（标准后缀为 **as** ）

c 编译程序体（标准后缀为 **ctcom** （基于 Itanium 的系统）或 **ccom** （PA-RISC 系统））

l 链接程序（标准后缀为 **ld** ）

p C 预处理器（标准后缀为 **cpp** ）

x 所有子进程。

-U*name* 取消在编译的预处理阶段之前定义的任何 *name* 的定义。

-v 启用详细模式，这将在标准错误中生成编译进程的逐步说明。

-V 显示当前编译程序和链接程序的版本号（如果执行了链接程序）。

-w 禁止显示警告消息。

-W*x*,*arg1*[,arg2...]

将逗号分隔的参数传递到子进程 *x* ，其中 *x* 可以假设其中一个值在 **-t** 选项以及 **d** （驱动程序）下列出，而不在 **x** （所有子进程）下列出。 **-W** 选项规范允许编译程序驱动程序识别附加的、特定于实现方法的选项。

仅限基于 **Itanium** 的系统选项

-z 切勿将任何内容绑定到地址零。该 *ld*(1) 选项允许运行时检测空指针。请参阅下面关于 指针的注释。（仅限基于 Itanium 的系统）

+dryrun 生成给定命令行的子进程信息，而不运行子进程。（仅限基于 Itanium 的系统）

+m[**d**] 该选项采用 *make*(1) 的格式输出用双引号 ("") 括起的包含文件的相关信息。使用 **+m** ，输出将转至 stdout。使用 **+md** ，输出将转至后缀为 **.d** 的文件中。该文件的目录和前缀与对象文件相同，会受 **-o** 选项的影响。

注释：指定 **-E** 可以禁止生成对象文件。在这种情况下，不生成任何预处理器输出。（仅限基于 Itanium 的系统）。

+M[**d**] 与 **+m** 相同，例外的是用 **<>** 和 括起的包含文件作为相关信息输出。（仅限基于 Itanium 的系统）。

+p 禁止所有不合时宜的结构。正常情况下，编译程序发出有关不合时宜的结构的警告；使用 **+p** 选项，编译程序将不编译含有不合时宜的结构的代码。（仅限基于 Itanium 的系统）。

+time 为编译子进程生成单独的计时信息。对于每个子进程，会为用户进程、系统调用和总处理时间生成估计时间（以秒为单位）。（仅限基于 Itanium 的系统）。

+uc 将“无格式的”char 数据类型视为无符号的 char。（使其超负荷和对其进行重整均不发生任何变化）使用该选项，可以从不合格（无格式）的 char 类型被视为无符号的 char（而不是缺省视为有符号的 char）的环境中转出应用程序。由于编译单元中所有不合格的 char 类型都将受到该选项的影响（包括定义外部和系统接口的头），所以有必要统一编译在一个程序中使用的所有接口。

+w 就所有可疑结构发出警告。如果不使用 **+w** 选项，编译程序将只发出有关确有问题的结构的警告（仅限基于 Itanium 的系统）。

+W*arg1*[,*arg2*,...,*argn*]
有选择地禁止显示任何指定的警告消息，其中 *arg1* 到 *argn* 是有效的编译程序警告消息号。（仅限基于 Itanium 的系统）。

+We*arg1*[,*arg2*,...,*argn*]
有选择地将任何指定的警告消息或将来的错误消息解释为错误。 *arg1* 到 *argn* 是有效的编译程序消息号。（仅限基于 Itanium 的系统）。

系统相关的选项

编译程序支持下列附加的操作系统相关选项。

+DD*data_model* 使用 *ILP32* 或 *LP64* 数据模型生成代码。 *data_model* 的定义值包括：

32 使用 ILP32 数据模型。整型、长整型和指针数据类型的大小都是 32 位。

64 使用 LP64 数据模型。整型数据类型的大小是 32 位，长整型和指针数据类型的大小是 64 位。将 __LP64__ 定义为预处理器。

+DS*model* 使用已调整为指定 *model* 的指令调度程序。如果未使用该选项，则编译程序将使用编译程序所用的体系结构的指令调度程序。

选项（其他）

其他未定义的任何选项都将生成一个标准错误警告。

其他参数会被视为 HP C 兼容的对象文件，这些对象文件通常由早期的 **cc** 运行而生成，或可能由 HP cc 兼容的例行程序的库所生成。这些文件以及指定的任何编译的结果将按照给定顺序链接起来，生成名为 **a.out** 的可执行程序。

指针

从技术上讲，访问一个空（零）指针的对象是非法的，但在过去许多系统都允许这样。假设下列条件可最大化代码的可移植性。如果硬件能够针对位置零的读取返回零（当以至少 8 位和 16 位进行访问时），除非存在 **-z** 标志，否则必须如此操作。如果试图访问位置零，则 **-z** 标志将请求生成 **SIGSEGV** 。即使未读取，也可能会将位置零的写入检测为错误。如果硬件无法确定位置零是初始化为零还是锁定为零，则硬件应将 **-z** 标志视为始终进行了设置。

外部语言环境影响

环境变量

LANG 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值 **C** （请参阅 *lang*(5) ）而不使用 **LANG** 。如果任一国际化变量包含无效设置，则 **C** 就会认为所有国际化变量都设置为 **C** 。请参阅 *environ*(5) 。

aCC_MAXERR 用于设置希望编译程序在终止编译前可以报告的错误的最大数量。（仅限基于 Itanium 的系统）。

SDKROOT 用作对工具集组件的所有引用的前缀，当使用非本地语言开发工具包或安装在其他位置的工具集时，必须设置该选项。有些工具集组件是编译程序驱动程序、编译程序应用程序、预处理器、链接程序和对象文件工具。

TARGETROOT 用作对目标集组件的所有引用的前缀，在使用非本地语言开发工具包时，也必须设置该选项。有些目标集组件是头文件、归档库和共享库。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

由编译程序本身生成的诊断信息旨在提供自述性信息。偶发性消息可能是由汇编程序或链接编辑器生成的。

如果在 **cc** 完成前出现任何错误，则将返回一个非零值。否则，将返回零。

相关内容

如果 **-s** 是随同上述任何选项一起指定的，则无论指定这些选项的顺序如何，都会忽略 **-s** 选项。

有关操作系统相关选项 **+DD***data_model* （用于 32 位或 64 位数据模型）和 **+DS***model* （用于指令调度程序）的信息，请参阅上面的 系统相关的选项小节。

文件

file.c	C 输入文件
file.i	以前预处理的 cc 输入文件
file.o	对象文件
file.so	共享库，在基于 Itanium 的系统上用 **-b** 创建
file.sl	共享库，在 PA-RISC 系统上用 **-b** 创建
a.out	链接的可执行输出文件
/var/tmp/*	编译程序使用的临时文件（基于 Itanium 的系统）
/var/tmp/ctm*	编译程序使用的临时文件（PA-RISC 系统）

/usr/ccs/bin/cc	C 驱动程序
/usr/ccs/bin/cc_bundled	C 驱动程序
/usr/ccs/lbin/ctcom	C 编译程序（基于 Itanium 的系统）
/usr/ccs/lbin/ccom	C 编译程序（PA-RISC 系统）
/usr/ccs/lbin/cpp	预处理器，用于汇编 **.s** 文件
/usr/lib/nls/msg/$LANG/aCC.cat	C 编译程序消息清单（基于 Itanium 的系统）
/usr/lib/nls/msg/$LANG/cc.cat	C 编译程序消息清单（PA-RISC 系统）
/usr/ccs/bin/as	汇编程序， *as*(1)
/usr/ccs/bin/ld	链接编辑器， *ld*(1)
/usr/ccs/lib/crt0.o	运行时启动（PA-RISC 系统）
/usr/include	**#include** 文件的标准目录

其他库

/usr/lib/hpux32/libc.so	标准 C 库，请参阅《 *HP-UX* 参考手册》的第 3 节。（基于 Itanium 的系统）
/usr/lib/hpux64/libc.so	标准 C 库，请参阅《 *HP-UX* 参考手册》的第 3 节。（基于 Itanium 的系统）
/usr/lib/libc.a	标准 C 库（归档版本），请参阅《 *HP-UX* 参考手册》的第 3 节。（PA-RISC 系统）
/usr/lib/libc.sl	标准 C 库（共享版本），请参阅《 *HP-UX* 参考手册》的第 3 节。（PA-RISC 系统）
/usr/lib/hpux32/libm.a	数学库（基于 Itanium 的系统）
/usr/lib/hpux64/libm.a	数学库（基于 Itanium 的系统）
/usr/lib/libm.a	数学库（PA-RISC 系统）
/usr/lib/hpux32/libdld.so	动态加载程序库（基于 Itanium 的系统）
/usr/lib/hpux64/libdld.so	动态加载程序库（基于 Itanium 的系统）
/usr/lib/libdld.sl	动态加载程序库（PA-RISC 系统）
/usr/lib/hpux32/dld.so	动态加载程序（基于 Itanium 的系统）
/usr/lib/hpux64/dld.so	动态加载程序（基于 Itanium 的系统）

/usr/lib/dld.so	动态加载程序（PA-RISC 系统）

另请参阅

联机帮助

可以使用缺省的 HTML 浏览器显示联机帮助，也可以通过 URL **file:/opt/ansic/html/$LANG/guide/index.htm** （基于 Itanium 的系统）或 **file:/opt/ansic/html/guide/$LANG/index.htm** （PA-RISC 系统）调用自己的 HTML 浏览器

其他可用主题包括：编译程序的经过定义的编译程序 Pragmas、浮动模式安装和实现方法等。

也可以通过以下网址获取信息： **http://www.hp.com/go/c**

配置处理和调试工具

gprof (1)	显示调用图形配置文件数据
monitor(3C)	准备执行配置文件
wdb(1)	C、C++ 和 Fortran 符号调试程序
gdb(1)	C、C++ 和 Fortran 符号调试程序
adb(1)	绝对调试程序

系统工具

as(1)	将汇编代码转换为计算机代码
cpp(1)	调用 C 语言预处理器
cc(1)	C 编译程序
ld(1)	调用链接编辑器

其他信息

strip(1)	从对象文件中去除符号和行号信息
crt0(3)	执行启动例程
end(3C)	程序中最后位置的符号
exit(2)	进程终止

教程和标准文档

《 American National Standard for Information Systems - Programming language C 》， ANS X3.159-1989.

请参阅《 HP C Online help 》。

名称

cc_bundled_pa： cc - 捆绑的 C 编译程序

概要

cc [*options*] *files*

说明

本联机帮助页介绍捆绑的 C 编译程序。有关符合 ANSI 的 HP-UX 的联机帮助页，请参阅 *cc*(1) （仅限联机）。

该 **cc** 接受以下几种参数作为 *files* ：

.c Suffix
: 其名称以 **.c** 结尾的参数被视为 C 源文件。每个文件将进行编译，得到的对象文件保留在带有对应基名的文件中，但该名称带有 **.o** 后缀而不是 **.c** 后缀。但是，如果在一个步骤中完成单个 C 文件的编译和链接，则将删除 **.o** 文件。

.s Suffix
: 其名称以 **.s** 结尾的参数被视为汇编源文件并进行汇编，从而为每个 **.s** 文件生成一个 **.o** 文件。

.i Suffix
: 其名称以 **.i** 结尾的参数假定为 **cpp** 的输出（请参阅下文的 **-P** 选项）。编译时不调用 **cpp** （请参阅 *cpp*(1) ）。每个对象文件保留在具有对应基名的文件，但其名称带有 **.o** 后缀而不是 **.i** 后缀。

-l*x* Form
: **-l***x* 形式的参数可使链接程序搜索库 **lib***x***.sl** 或 **lib***x***.a** ，以解析当前未解析的外部引用。由于在遇到库名时对库进行搜索，因此 **-l** 的放置位置非常重要。如果文件包含未解析的外部引用，则必须在命令行上将包含定义的库放在文件 之后。有关详细信息，请参阅 *ld*(1) 。

-l:lib*x***.***suffix* Form
: **-l:lib***x***.***suffix* 形式的参数可使链接程序搜索库 **lib***x***.sl** 或 **lib***x***.a** （取决于 *suffix* ），以解析当前未解析的外部引用。它与 **-l** 选项类似，但例外的是 **-Wl,-a** 选项的当前状态不重要。

Other Suffixes
: 其他所有参数（如其名称以 **.o** 或 **.a** 结尾的参数）被视为将在链接操作中包括的可重定位对象文件。

参数和选项可以通过 **CCOPTS** 环境变量以及命令行传递给编译程序。编译程序读取 **CCOPTS** 的值，并将这些选项分为两个集合；在竖线 (|) 之前出现的选项和在竖线之后出现的选项。第一个选项集放置在 **cc** 的任何命令行参数之前；第二个参数集放置在 **cc** 的命令行参数之后。如果没有竖线，所有选项均放在命令行参数之前。例如（在 *sh*(1) 表示法中），

```
CCOPTS="-v | -lmalloc"
export CCOPTS
cc -w prog.c
```

等效于

```
cc -v -w prog.c -lmalloc
```

如果已设置， **TMPDIR** 环境变量指定将由编译程序用于临时文件的目录，它覆盖缺省目录 **/var/tmp** 。

选项

以下选项是绑定的 C 编译程序可识别的全部选项。

-c 禁止编译的链接编辑阶段，并强制为每个 **.c** 文件生成一个对象 (**.o**) 文件（即使仅编译一个程序）。执行之前必须链接 C 程序生成的对象文件。

-C 禁止预处理器删除 C 式样的注释。有关详细信息，请参阅 *cpp*(1) 。

-D*name*=*def*
-D*name* 为预处理器定义 *name* ，就像是通过“#define”一样。有关详细信息，请参阅 *cpp*(1) 。

-E 在命名的 C 或汇编文件上仅运行 **cpp** ，然后将结果发送到标准输出。

-I*dir* 更改预处理器用来查找包含文件的算法，以附加搜索目录 *dir* 。有关详细信息，请参阅 *cpp*(1) 。

-l*x* 请参阅说明部分。

-L *dir* 更改链接程序用来搜索 **lib***x***.sl** 或 **lib***x***.a** 的算法。 **-L** 选项可使 **cc** 在缺省位置中搜索之前先搜索 *dir* 。有关详细信息，请参阅 *ld*(1) 。

-o*outfile* 命名链接程序 *outfile* 中的输出文件。缺省名称是 **a.out** 。

-P 在命名的 C 文件上仅运行 **cpp** ，并将结果保留在带 **.i** 后缀的相应文件上。 **-P** 选项也传递给 **cpp** 。

+R*num* 仅允许前 *num* 个 **register** 变量实际具有 **register** 类。在寄存器分配器发出以下消息时使用该选项：

out of general registers

-s 可使链接程序的输出删除符号表信息。有关详细信息，请参阅 *strip*(1) 。如果使用该选项，将防止对得到的程序使用符号调试程序。有关详细信息，请参阅 *ld*(1) 。

-S 编译命名的 C 文件，并将汇编语言输出保留在带 **.s** 后缀的相应文件上。

-t*x*,*name* 将子进程 *x* 替换为 *name* ，其中 *x* 是一个或多个指示子进程的标识符集。该选项在两种模式下工作：1) 如果 *x* 是单个标识符， *name* 表示新子进程的完整的路径名；2) 如果 *x* 是标识符集， *name* 表示前缀，标准后缀将连接到该前缀来构成新子进程的完整路径名。

x 可以采用下面的一个或多个值：

p 预处理器（标准后缀是 **cpp** ）
c 编译程序（标准后缀是 **ccom** ）
a 汇编程序（标准后缀是 **as** ）
l 链接程序（标准后缀是 **ld** ）

-U*name* 删除预处理器中 *name* 的任何初始定义。有关详细信息，请参阅 *cpp*(1) 。

-v 启用详细模式，它在标准错误上生成编译进程的分布说明。

-V 可使每个调用的子进程将其标准信息输出到 stdout。

-w 禁止警告消息。

-W*x***,***arglist* 将 *arglist* 中的逗点分隔参数传递给子进程 *x* 。如果指定了 **-W** 选项，则允许编译程序驱动程序识别更多针对实现的选项。例如，

-Wl,-a,archive

可使链接程序与归档库链接，而不是与共享库链接。有关详细信息，请参阅 *ld*(1) 。

x 可以采用下列值之一：

p 预处理器
a 汇编程序
l 链接程序

遇到任何其他选项，均会生成一个警告并发送到/写入标准错误。

假定其他参数是与 C 兼容的对象程序（它们通常由运行早期的 **cc** 生成），或者可能是兼容 C 的例程库。这些程序与指定的任何编译的结果一起进行链接（按给定的顺序），生成名为 **a.out** 的可执行程序。

外部语言环境影响

环境变量

LANG 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省的 **C** （请参阅 *lang*(5) ）。如果任一国际化变量包含无效设置，则 **cc** 就会认为所有国际化变量都设置为 **C** 。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

C 本身生成的诊断信息应该是自述性消息。有时，消息会由预处理器、编译程序或链接编辑器生成。

如果在完成 **cc** 之前出错，则返回非零的值。否则，将返回零。

举例

以下示例编译 C 文件 **prog.c** 以创建 **prog.o** 文件，然后调用 **ld** 链接编辑器将 **prog.o** 和 **procedure.o** 与 **/usr/ccs/lib/crt0.o** 中的 C 启动例程和 C 库 **libc.sl** 或 **libc.a** 中的库例程相链接。得到的可执行程序放入文件 **prog**：

cc prog.c procedure.o -o prog

警告

cc 不识别的选项不会传递给链接编辑器。选项 **-Wl,***arg* 可用于将这样的任何选项传递给链接编辑器。

缺省情况下，C 程序的返回值是完全随机的。要返回特定值，唯一的两种方式是显式调用 **exit()** （请参阅 *exit*(2)）或通过 **return** *expression***;** 结构退出函数 **main()** 。

文件

file.c	输入文件
file.o	对象文件
a.out	链接的可执行输出文件
/var/tmp/ctm*	编译程序使用的临时文件
/usr/ccs/bin/as	编译程序（请参阅 *as*(1)）
/usr/ccs/bin/ld	链接编辑器（请参阅 *ld*(1)）
/usr/ccs/lib/crt0.o	运行时启动
/usr/lib/libc.a	标准 C 库（归档版本），请参阅《HP-UX 参考手册》第 (3) 节
/usr/lib/libc.sl	标准 C 库（共享版本），请参阅《HP-UX 参考手册》第 (3) 节
/usr/include	**#include** 文件的标准目录

绑定的 C 编译程序文件

/usr/ccs/bin/cc	C 驱动程序
/usr/ccs/lbin/ccom	C 编译程序
/usr/lib/nls/msg/$LANG/cc.cat	C 编译程序消息清单
/usr/ccs/lbin/cpp	C 预处理器

另请参阅

系统工具

as(1)	将汇编代码转换为机器代码。
cpp(1)	调用 C 语言预处理器。
ld(1)	调用链接编辑器。
cc(1)	HP-UX 上符合 ANSI 的 C 编译程序。

其他信息

matherr(3M)	捕获数学错误。
fpgetround(3M)	浮点模式控制函数。
strip(1)	从对象文件中删除符号和行号信息。
crt0(3)	执行启动例程。
end(3C)	程序最后位置的符号。
exit(2)	进程的终止。

教程和标准文档

B. W. Kernighan 和 D. M. Ritchie，《The C Programming Language》，Prentice-Hall，1978 年。

名称

cd - 更改工作目录

概要

cd [*directory*]

说明

如果未指定 *directory* ，则使用 Shell 参数 **HOME** 的值作为新的工作目录。如果 *directory* 指定了以 **/** 、 **.** 或 **..** 开头的完整路径，则 *directory* 将成为新的工作目录。如果两种情况都不适用，则 **cd** 尝试查找与 **CDPATH** Shell 变量指定的路径之一有关的指定目录。 **CDPATH** 与 **PATH** Shell 变量具有相同语法和相近语义。 **cd** 必须在 *directory* 中具有执行（搜索）权限。

cd 只能作为 Shell 内置命令存在，这是因为无论何时执行一条命令，都会创建一个进程，而如果作为常规的系统命令写入和处理 **cd** ，则会变为无效。而且，不同的 Shell 提供作为内置实用程序的 **cd** 的不同实现。此处描述的 **cd** 功能可能得不到所有 Shell 的支持。有关各 Shell 的区别，请参考相应的手册条目。

如果在一个子 Shell 中或单独的实用程序执行环境下调用 **cd** ，例如：

find . -type d -exec cd {}; -exec foo {};

（对可访问的目录调用 **foo** ） **cd** 不会影响调用方的环境的当前目录。 **cd** 作为独立命令使用的另一用途是获得命令的退出状态。

外部语言环境影响

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

环境变量

下列环境变量会影响 **cd** 的执行：

HOME 主目录的名称，在未指定任何目录操作数的情况下使用。

CDPATH 表示目录的由冒号分隔的路径名列表。如果目录操作数不以斜线 (/) 字符开头，且第一个组成部分不是 . 或 ..，则 **cd** 将按列出的顺序搜索相对于 **CDPATH** 变量中命名的每一目录的 *directory* 。新的工作目录被设置为找到的第一个匹配目录。如果目录路径名为空字符串，则表示当前目录。如果未设置 **CDPATH** ，则会将视其为空字符串。

举例

在文件系统的任意位置将当前工作目录更改到 **HOME** 目录：

cd

更改到位于当前目录中的新的当前工作目录 **foo** ：

cd foo

或者

cd ./foo

更改到位于当前目录的父目录中的目录 **foobar**：

cd ../foobar

更改到其绝对路径名为 **/usr/local/lib/work.files** 的目录:

cd /usr/local/lib/work.files

更改到相对于主目录的目录 **proj1/schedule/staffing/proposals**：

cd $HOME/proj1/schedule/staffing/proposals

返回值

完成后， **cd** 退出时返回下列值之一:

0	已成功更改了目录。
>0	发生了错误。工作目录保持不变。

另请参阅

csh(1)、 pwd(1)、 ksh(1)、 sh-posix(1)、 sh(1)、 chdir(2)。

符合的标准

cd：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

cdc - 更改 SCCS delta 版的 delta 版注释

概要

cdc -r *SID* [**-m**[*mrlist*]] [**-y**[*comment*]] *files*

说明

cdc 命令更改每个命名的 SCCS 文件的 **-r** 选项指定的 *SID* 的 “**delta** 版注释”。

“**delta** 注释”定义成修改请求 (MR) 和注释信息，它们通常通过 *delta*(1) 命令（ -m 和 **-y** 选项）指定。

如果命名了目录， **cdc** 会像目录中的每个文件都指定为命名文件那样进行操作，不同的只是将以静默方式忽略非 SCCS 文件（路径名的最后一部分不以 **s.** 开头）和不可读的文件。如果给定了 **-** 的名称，则将读取标准输入（请参阅 警告）；标准输入的每一行将用作要处理的 SCCS 文件的名称。

选项

cdc 的参数（可以按任意顺序出现）由 *option* 参数和文件名组成。

所述的全部 *option* 参数均独立地应用于每个命名的文件：

-r*SID* 用于指定要更改其 delta 版注释的 delta 版的 *S*CCS *ID*entification (SID) 字符串。

-m[*mrlist*] 如果 SCCS 文件已设置 **v** 选项（请参阅 *admin*(1) ），则 可能提供要在 **-r** 选项所指定的 *SID* 的 delta 版注释中添加和（或）删除的 MR 编号的列表。空 MR 列表不起作用。

MR 条目按照与 *delta*(1) 相同的方式添加到 MR列表中。要删除 MR ，请在 MR 编号前添加字符 **!** （请参阅 举例）。如果要删除的 MR 当前位于 MR列表中，则会将其删除并移到“注释”行中。所有已删除的 MR的列表将放入 delta 版注释的注释部分，并且在前面添加声明它们已删除的命令行。

如果未使用 **-m** 并且标准输入是终端，则将在读取标准输入之前在标准输出上发出提示 **MRs?** ；如果标准输入不是终端；则不发出提示。 **MRs?** 提示始终位于 **comments?** 提示之前（请参阅 **-y** 选项）。

列表中的 MR 由空白字符和（或）制表符分隔。非转义的换行符将终止 MR列表。

请注意，如果 **v** 选项带有值（请参阅 *admin*(1) ），则会将其视为用于验证 MR 编号正确性的程序（或 Shell 过程）的名称。如果 MR 编号验证程序返回非零退出状态， **cdc** 将终止，而 delta 版注释保持不变。

-y[*comment*] 用于替换 *comment* 的任意文本或者由 **-r** 选项指定的 delta 版的现有 *comment* 。先前的注释将保留下来，并在前面添加声明它们已更改的命令行。空 *comment* 不起作用。

如果未指定 **-y** 并且标准输入是终端，则将在读取标准输入之前在标准输出上发出提示 **comments?** ；如果标准输入不是终端；则不发出提示。非转义的换行符将终止 *comment* 文本。

修改 SCCS 文件所必需的的确切权限记录在 *get*(1) 中。简单而言，它们是：

- 如果您创建了 delta 版，则可以更改其 delta 版注释，或者
- 如果您拥有该文件和目录，则可以修改 delta 版注释。

外部语言环境影响

环境变量

LANG 用于确定显示消息的语言。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

可使用 *sccshelp*(1) 获得进一步说明。

举例

将 **bl78-12345** 和 **bl79-00001** 添加到 MR 列表，将 **bl77-54321** 从 MR 列表中删除，并且将注释 **trouble** 添加到 **s.file** 的 delta 版 1.6：

```
cdc -r1.6 -m"bl78-12345 !bl77-54321 bl79-00001" -ytrouble s.file
```

以下命令执行相同的操作：

```
cdc -r1.6 s.file
MRs? !bl77-54321 bl78-12345 bl79-00001
comments? trouble
```

警告

如果通过标准输入将 SCCS 文件名提供给 **cdc** 命令（命令行上的 - ），则还必须使用 **-m** 和 **-y** 选项。

文件

x-file 请参阅 *delta*(1) 。

z-file 请参阅 *delta*(1) 。

另请参阅

admin(1)、 delta(1)、 get(1)、 sccshelp(1)、 prs(1)、 sccsfile(4)、 rcsfile(4)、 acl(5)、rcsintro(5)。

名称

chacl - 添加、修改、删除、复制或汇总文件的访问控制列表 (ACL)

概要

/usr/bin/chacl *acl file*

chacl -r *acl file*

chacl -d *aclpatt file*

chacl -f *fromfile tofile*

chacl - [**z** | **Z** | **F**] *file*

说明

通过 **chacl** ，用户可以授予或限制对特定用户和（或）组的文件访问，从而扩展了 *chmod*(1) 的功能。传统文件访问权限在创建文件时设置，它们授予或限制文件的所有者、组和其他用户的访问权。这些文件访问权限（如 **rwxrw-r--** ）映射到三个基本访问控制列表条目：一个条目用于文件的所有者 (***u*.%,** *mode*)，一个条目用于文件的组 (**%.***g*, *mode*) ，一个条目用于其他用户 (**%.%,** *mode*)。

通过 **chacl** ，用户可以指定多达 13 个附加的权限集（称作可选访问控制列表 (ACL) 条目），这些权限集存储在文件的访问控制列表中。

要使用 *chacl* ，所有者（或超级用户）需构建一个 *acl* ，即与一个或多个文件相关联的一组 *(user.group, mode)* 映射。特定用户和组可以按名称或编号引用；任意用户 (*u*) 、组 (*g*) 或这两者均可以通过 **%** 符号（表示 任意用户或组）来引用。 **@** 符号指定文件的所有者或组。

读取、写入和执行（或搜索） (**rwx**) *modes* 与 *chmod* 所使用的模式完全相同；符号运算符 (*op*) 添加 (**+**) 、删除 (**-**) 或设置 (**=**) 访问权限。如果 *acl* 包含空格或特殊字符，则应该引用整个 acl。虽然可以通过两种方式构建 *acl* （ *acl*(5) 对此有详尽的介绍），但建议采用以下语法：

entry [, *entry*] ...

其中 *entry* 的语法是

u.g op mode[*op mode*] ...

缺省情况下， **chacl** 会修改现有的 ACL它将在现有的 ACL 条目中添加 ACL 条目或修改访问权限。如果 *acl* 包含已经与文件相关联的 ACL 条目，则该条目的模式位将更改为给定的新值，或者由指定的运算符进行修改。如果文件的 ACL 尚未包含指定的条目，则将添加该 ACL 条目。 **chacl** 也可删除文件的所有访问权。如果为其给定空 *acl* 参数，则意味着“无访问权”（使用 **-r** 选项时）或“无更改”。

若需语法摘要，请运行不带参数的 **chacl** 。

如果 *file* 指定为 **-** ， **chacl** 将从标准输入中读取。

选项

chacl 可识别下列选项：

-r 将旧的 ACL替换为给定的 ACL 。所有可选 ACL 条目将首先从指定文件的 ACL中删除，将其基本权限设置为零，然后应用新的 ACL 。如果 *acl* 不包含文件的所有者 (***u.%***)、组 (***%.g***) 或其他用户 (***%.%***) 的条目，则该基本 ACL 条目的模式将设置为零（无访问权）。该命令会影响文件的所有 ACL 条目，但不更改文件的所有者或组 ID 。

在 *chmod*(1) 中，“修改”和“替换”操作通过语法（字符串或八进制值）来区分。由于 ACL 包含可变数量的条目，因此没有 ACL 的必然的结果。因此， **chacl** 将在缺省情况下修改特定的条目，并选择性地替换所有条目。

-d 从所有指定文件上的 ACL 中删除指定的条目。 *aclpatt* 参数可以是确切的 ACL ，也可以是 ACL 模式（请参阅 *acl*(5) ）。只有从 ACL 中删除条目时， **chacl -d** 才会更新每个文件的 ACL 。

如果您尝试从任意文件中删除基本 ACL 条目，该条目将保留下来，但其访问模式将设置为零（无访问权）。如果尝试从文件中删除不存在的 ACL 条目（即，如果 ACL 条目模式不匹配 ACL 条目）， **chacl** 将向您通知错误，继续执行，最后返回非零。

-f *fromfile tofile* 将 ACL 从 *fromfile* 复制到指定的 *tofile* ，并在必要时转移所有权（请参阅 *acl*(5) 、 *chown*(2) 或 *chownacl*(3C) ）。 *fromfile* 可以是 **-** ，它表示标准输入。

该选项暗含了 **-r** 选项的作用。如果 *fromfile* 的所有者和组与 *tofile* 的所有者和组相同，则 **chacl -f** 等于：

chacl -r ‘lsacl fromfile‘ tofile ...

要复制 ACL 而不转移所有权，建议使用以上命令，而不是 **chacl -f** 。

-z 删除（“zap”）指定文件的 ACL中的所有可选条目，仅保留基本条目。

-Z 删除（“zap”）指定文件的 ACL中的所有可选条目，并将所有基本条目中的访问模式设置为零（无访问权）。它等效于将旧的 ACL 替换为空 ACL：

chacl -r ” file ...

或者使用 *chmod*(1) ，它的副作用是会删除可选条目：

chmod 0 *file* ...

-F 将可选的 ACL 条目合并（“fold”）到基础 ACL 条目中。如有必要，将更改基本 ACL 条目的权限位，以反映调用方对文件的有效访问权限；所有可选条目（如果有）均会被删除。

对于普通用户，只能更改所有者基础 ACL 条目的访问模式。与 **getaccess** 不同，对于只读文件系统上的文件或正在执行的共享文本程序，将不关闭写入位（请参阅 *getaccess*(1) ）。

对于超级用户，只有当文件不是常规文件，或者执行位尚未在基本 ACL 条目模式下设置，但在可选的 ACL 条目模式下设置时，才可能只更改所有者基本 ACL 条目中的执行模式位。

也可以从文件内的字符串中获取 *acl* ：

chacl 'cat file' *files* ...

如果使用 *acl* 中的 @ 表示“文件所有者或组”，可能会导致 **chacl** 运行速度减慢，这是因为它必须重新分析每个文件的 ACL （使用 **-d** 选项时除外）。

外部语言环境影响

环境变量

LANG 用于确定显示消息的语言。

如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省的“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **chacl** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

返回值

如果 **chacl** 成功，返回值将为零。

如果 **chacl** 在更改任何文件的 ACL 之前遇到错误，它会将错误消息输出到标准错误，并返回 1。这些错误包括调用无效、 *acl* (*aclpatt*) 语法无效、给定用户名或组名未知或者无法用 **-f** 选项从 *fromfile* 中获取 ACL 。

如果 **chacl** 无法执行请求的操作，它将向标准错误输出错误消息，继续执行，并在稍后返回 2。其中包括文件不存在、文件的 ACL 无法更改、得到的 ACL 条目超过允许值或者尝试删除不存在的 ACL 条目等情况。

举例

以下命令为任意组中的用户 **jpc** 添加读取权限，并删除文件组中任何用户的写入权限（对于文件 **x** 和 **y** ）。

chacl "jpc.%+r, %.@-w" x y

该命令将作为标准输入打开的文件和文件 **test** 上的 ACL 替换为只允许文件所有者具有读写访问权的 ACL。

chacl -r '(@.%,rw-)' - test

从文件 **myfile** 中删除组 13 中用户 165 的特定访问权限（如果有）。请注意，这不同于添加限制该用户和组的访问权的 ACL 条目。用户的结果访问权限取决于 ACL 中剩余的条目。该命令还删除用户 **jpc** 的所有已打开读取位的条目（星号可以用作用户、组或访问模式的 ACL 模式中的通配符）：

chacl -d '165.13, jpc.*+r' myfile

将 ACL 从 **oldfile** 复制到 **slow/hare** 和 **fast/tortoise** 。

chacl -f oldfile slow/hare fast/tortoise

删除作为标准输入打开的文件上的可选 ACL 条目（如果有）。

chacl -z -

拒绝对当前目录中其名称以 **a** 、 **b** 或 **c** 开头的所有文件的所有访问:

chacl -Z [a-c]*

将文件 (**fun.stuff**) 的可选 ACL 条目合并到基本 ACL 条目中:

chacl -F fun.stuff

警告

ACL 字符串所包含的唯一条目不可超过 16 个（即使将 @ 符号转换成用户名或组名并合并冗余条目后，可能会使某些文件的条目少于 16 个）。

相关内容

如果目标文件驻留在一个不支持 ACL 的文件系统上， **chacl** 将会失败。

NFS

远程文件上仅支持 **-F** 选项。

作者

chacl 由 HP 开发。

另请参阅

chmod(1)、 getaccess(1)、 lsacl(1)、 getacl(2)、 setacl(2)、 acl(5)、 glossary(9)。

名称

chatr - 更改程序的内部属性

概要

备注

对于基于 Itanium® 的系统，请参阅 *chatr_ia*(1) 。

对于 PA-RISC 系统，请参阅 *chatr_pa*(1) 。

使用 **uname** 命令确定您的系统类型。在基于 Itanium 的系统上， **uname -m** 返回 **ia64**。所有其他值表示 PA-RISC 系统。

另请参阅

chatr_ia(1)、 chatr_pa(1)、 uname(1)。

名称

chatr_ia: chatr - 更改程序的内部属性

概要

格式 1：适用于具有单个文本段和单个数据段的文件

chatr [**-s**] [**-z**|**Z**] [**-l** *library*] [**-B** *mode*] [**+as** *mode*] [**+b** *flag*] [**+cd** *flag*] [**+ci** *flag*] [**+dbg** *flag*] [**+es** *flag*] [**+gst** *flag*] [**+gst-size** *size*] [**+id** *flag*] [**+k** *flag*] [**+l** *library*]
[**+md** *flag*] [**+mergeseg** *flag*] [**+mi** *flag*] [**+o** *flag*] [**+pd** *size*] [**+pi** *size*] [**+s** *flag*] [**+z** *flag*] [**+I** *flag*] *file ...*

格式 2：适用于采用显式规范的段

chatr {**+sa** *address* | **+sall** | **+si** *index*} [**-s**] [**-B** *mode*] [**+c** *flag*] [**+dz** *flag*] [**+k** *flag*]
[**+m** *flag*] [**+mergeseg** *flag*] [**+p** *size*] [**+r** *flag*] [**+s** *flag*] [**+z** *flag*] [**+I** *flag*] *file ...*

说明

chatr 可用于为 32 位和 64 位 ELF 文件更改程序的内部属性。

可以使用两种句法格式调用 **chatr** 。

- “格式 **1**”允许方便地操作只有单个文本段和单个数据段的普通文件。
- “格式 **2**”允许修改采用显式规范的段。

完成后，如果没有指定 **-s** ， **chatr** 会将文件的旧值和新值均输出到标准输出。

+pd 和 **+pi** 选项只用于提示虚拟内存页面大小。实际页面大小可能有所不同。在某些情况下， **L** 的页面大小提示可能会提高性能，这取决于应用程序的特定内存要求。

有些应用程序的性能可能会受益于静态分支预测，而其他应用程序则不能。 **+r** 选项提供了使用或避免使用该功能的提示。

+gst 和相关选项使用可提高导出符号搜索功能的全局符号表，从而实现了性能上的改进。有关详细信息，请参阅 *dld.so*(5) 和《HP-UX Linker and Libraries Online User Guide》。

要使用格式 2，请先指定要通过 *address* （使用 **+sa** 选项）或 *index* （使用 **+si** 选项）修改的段，或指定所有段（使用 **+sall** 选项）。然后使用 **+c** 、 **+m** 、 **+r** 、 **+s** 或 **+z** 选项修改段属性。只要用 **+sa** *address* 或 **+si** index 选项指定了每个段，并在这些选项后跟修改选项，就可以在命令行上包括多个段。

选项

-l *library* 如果提供了目录路径列表，则表示指定的共享库容易受运行时路径查找的影响（请参阅 **+s** 和 **+b**）。

-s 以静默方式执行其操作。

-z 启用空指针的运行时取消引用来产生 SIGSEGV 信号。（它是对 **-Z** 选项的补充）。

-B *mode* 选择程序（使用共享库）的运行时绑定行为模式。必须指定一种绑定模式 **immediate** 或 **deferred** 。有关绑定模式的说明，请参阅《HP-UX Linker and Libraries Online User Guide》。

-Z 禁用空指针的运行时取消引用。（它是对 **-z** 选项的补充）。

+as *mode* 控制将由内核使用的地址空间模式。可能的模式值为 **default** 、 **share_magic** 、 **exec_magic** 、 **shmem_magic** 和 **mpas** 。缺省值当前等同于 **share_magic** 。为了将模式设置为缺省值之外的任何值，应事先使用 **-N** 编译程序选项构建二进制文件，以确保文本段和数据段是相邻的。

+b *flag* 控制在构建程序（如果有）时存储的嵌入式路径列表是否可用于查找该程序所需的共享库。两个标志值 **enable** 和 **disable** 分别用于启用和禁用嵌入式路径列表。但不能对 ELF 文件使用 **disable**，否则将发出一条警告消息。请参阅 **+s** 选项。可以使用 **+b** 选项为过滤器库启用嵌入式路径。

+c *flag* （仅限格式 2。）启用或禁用指定段的代码位。如果已启用，则对于 **chatr** 输出中列出的段，代码位将用 **c** 标志表示。

+cd *flag* 启用或禁用文件的数据段的代码位。如果已启用，则对于 **chatr** 输出中列出的段，代码位将用 **c** 标志表示。

+ci *flag* 启用或禁用文件的文本段的代码位。如果已启用，则对于 **chatr** 输出中列出的段，代码位将用 **c** 标志表示。

+dbg *flag* 启用或禁用运行程序的功能，并在运行之后添加一个调试程序，然后在其相关共享库中设置断点。

+dz *flag* （仅限格式 2。）启用或禁用动态分配段（如堆栈或堆）的惰性交换分配。

+es *flag* 用 *flag* 值 **enable** 和 **disable** 控制从堆栈执行用户代码的能力。有关安全问题的其他信息，请参阅下文 限制对堆栈的执行权限一节。

+gst *flag* 控制是否将全局符号表散列机制用于查找符号导入（或导出）条目的值。两个标志值 **enable** 和 **disable** 分别用于启用和禁用全局符号表散列机制。缺省值为 **disable** 。

+gstsize *size* 使用全局符号表散列机制请求特定的散列阵列 *size* 。该值介于 1 和 **MAXINT** 之间。缺省值为 1103。将该选项与 **+gst enable** 配合使用。对文件使用该选项的工作方式与 **+gst** 选项类似。

+id *flag* 控制是否为数据段优先选择物理内存。该选项仅对 ccNUMA （Cache Coherent Non-Uniform Memory Architecture，高速缓存相关的非一致性内存体系结构）系统很重要。标志值可以是启用或禁用。如果启用，数据段将使用交错式内存。如果禁用（缺省设置），数据段将使用单元本地内存。该行为将跨 **fork()** 而不是 **exec()** 继承。

有关 ccNUMA 的详细信息，请参阅 *pstat_getlocality*(2) 。

+k *flag* 请求内核辅助的分支预测。标志 **enable** 和 **disable** 分别用于打开和关闭该请求。

+l *library* 如果提供了目录路径列表，则表示指定的共享库不受运行时路径查找的影响（请参阅 **+s** 和 **+b**）。

+m *flag* （仅限格式 2。）启用或禁用指定段的修改位。如果已启用，则对于 **chatr** 输出中列出的段，修改位将用 **m** 标志表示。

+md *flag* 启用或禁用文件的数据段的修改位。如果已启用，则对于 **chatr** 输出中列出的段，修改位将用 **m** 标志表示。

+mergeseg *flag* 启用或禁用共享库段合并功能。如果启用，则在程序启动时加载的所有共享库数据段将合并为一个块。每个动态加载库的数据段也将与其相关库的数据段合并。合并这些数据段后，内核可以使用更大的页面表条目，从而可以提高运行时性能。

+mi *flag* 启用或禁用文件的文本段的修改位。如果已启用，则对于 **chatr** 输出中列出的段，修改位将用 **m** 标志表示。

+o *flag* 启用或禁用 **DF_ORIGIN** 标志，以控制在计算工作目录绝对路径过程中 **$ORIGIN** 的使用情况。启用该标志，将指示在第一次加载父模块（对象模块、共享库或可执行文件）时计算当前工作目录的绝对路径。加载程序随后将该路径用于 **$ORIGIN** 的所有实例。然后，加载程序将该路径用于相关库中 **$ORIGIN** 的所有实例。

如果没有 **$ORIGIN** 的实例，则应该禁用 **DF_ORIGIN** 标志，以避免计算绝对路径。缺省情况下，如果 **$ORIGIN** 不存在，则将禁用 **DF_ORIGIN** 标志。

+p *size* （仅限格式 2。）为指定段设置页面大小。

+pd *size* 请求应该用于数据的特定虚拟内存页面大小。支持的大小包括 **4K** 、 **16K** 、 **64K** 、 **256K** 、 **1M** 、 **4M** 、 **16M** 、 **64M** 、 **256M** 、 **1G** 、 **4G D** 和 **L** 。 **D** 大小将导致使用缺省页面大小。**L** 大小将导致使用可用的最大页面大小。如果无法实现请求的大小，则实际页面大小可能有所不同。

+pi *size* 请求应该用于文本（指令）的特定虚拟内存页面大小。有关其他信息，请参阅 **+pd** 选项。

+r *flag* 当执行该程序时请求静态分支预测。标志 **enable** 和 **disable** 分别用于打开和关闭该请求。如果已启用，则对于 **chatr** 输出中列出的段，请求将用 **r** 标志表示。

+s *flag* 控制用 **LD_LIBRARY_PATH** 和 **SHLIB_PATH** 环境变量指定的目录路径列表是否可用于查找程序所需的共享库。两个标志值 **enable** 和 **disable** 分别用于启用和禁用环境变量。如果同时使用了 **+s** 和 **+b** ，则它们在命令行上的相对顺序将指示首先搜索哪个路径列表。请参阅 **+b** 选项。

+sa *address* （仅限格式 2。）使用一个地址为一组属性修改指定一个段。

+sall （仅限格式 2。）为一组属性修改使用文件中的所有段。

+si *index* （仅限格式 2。）使用一个段索引号为一组属性修改指定一个段。

+z *flag* 对所有数据段（使用格式 1）或特定段（使用格式 2）启用或禁用惰性交换。标志 **enable** 和 **disable** 分别用于打开或关闭该请求。不可用于非数据段。

+I *flag* 启用或禁用通过 **/opt/langtools/bin/caliper** 可实现的动态配置。如果启用，动态加载程序（请参阅 *dld.so*(5) ）将在程序执行时自动调用 **caliper** ，以收集配置文件信息。

限制对堆栈的执行权限

侵占系统的一种常用方法是，恶意使某程序堆栈发生缓冲区溢出，例如将特别长的、精选命令行参数传递到不需要它们的特权程序。怀有恶意的无权限用户可以利用这种方法欺骗特权程序，使其为自己启动超级用户 Shell，或执行类似的未经授权的操作。

降低此类攻击风险的一个简单而非常有效的方法是，从程序的堆栈页中删除执行权限。这将在不降低系统性能的情况下增强系统安全性，并且对绝大多数的合法应用程序不产生负面影响。本节介绍的更改仅影响极少数的程序，这些程序会试图执行（或被欺骗执行）程序堆栈上的指令。

如果为某个程序启用了本节所述的堆栈保护功能，但该程序试图从堆栈中执行代码，则 HP-UX 内核将终止该程序，同时发出 **SIGKILL** 信号，还显示一条要求查阅本手册页面部分的消息，同时在系统消息日志中记录一条错误消息（使用 **dmesg** 可查看该错误消息）。内核记录的消息为：

> **WARNING: UID # may have attempted a buffer overflow attack. PID # (program_name) has been terminated. See the '+es enable' option of chatr(1).**

如果遇到上述消息之一，请与该程序所有者联系，以便确定该程序是否正在合法地从堆栈中执行代码。如果是，则可以使用下述两种方法或其中一种方法，使程序重新正常运行。如果该程序正在从堆栈中非法地执行代码，则应该猜想是否存在恶意活动并采取适当的操作。

HP-UX 提供了两种允许从程序堆栈合法执行代码的方法。组合使用这两种方法，有助于在安全性和兼容性之间实现站点特定的利弊权衡。

第一种方法是使用 **chatr** 的 **+es** 选项，从而影响单个程序。这种方法通常用于指定某个二进制文件必须能够从其堆栈执行，而无论系统缺省设置如何。这样就允许使用限制性的系统缺省值，同时不禁止合法程序在它们的堆栈上执行代码。从理论上讲，该选项应由程序提供者设置（如果需要），以便将程序安装者进行手动干预的的需要降到最低程度。

另一种方法是设置内核可调参数 **executable_stack**，从而针对堆栈是否可执行设置一个系统级的缺省值。用 **sam**（请参阅 *sam*(1M)）将 **executable_stack** 参数设置为 1（一），将指示 HP-UX 内核允许程序在程序堆栈上执行。如果与较旧版本的兼容性比安全性更为重要，则使用该设置。如果安全性比兼容性更为重要，那么合适的做法是将 **executable_stack** 参数设置为 0（零），此为建议设置。该设置会显著增强系统安全性，同时对合法应用程序的负面影响（如果有）也最小。

组合使用这些设置能够适合许多应用程序的需要。例如，将 **executable_stack** 设置为 0 后，您可能会发现有一个或两个关键应用程序无法正常工作，这是因为它们需要从堆栈合法地执行。使用自修改代码的模拟程序或解释程序等程序，就属于您可能会遇到的上述程序实例。要获得使用限制性系统缺省值的安全性优点，同时仍然保证这些特定应用程序能够正常运行，请将 **executable_stack** 设置为 0，然后对需要从堆栈执行代码的特定二进制程序运行 **chatr +es enable**。由于这些二进制代码会输出要求查阅本手册页的错误消息，因此它们在执行时，很容易就能识别出来。

executable_stack 的可能设置如下所示：

executable_stack = 0

（缺省值）设置为 0（缺省值）会导致堆栈无法执行，因此从安全角度考虑，强烈建议采用此设置。

executable_stack = 1

设置为 1 会导致所有程序堆栈都可以执行，从兼容性角度考虑，此设置是最可靠的，但对于此参数而言，则是安全性最低的设置。

executable_stack = 2

设置为 2 等同于设置为 0，不过该设置会发出非致命性警告，而不是终止试图从其堆栈执行的进程。使用该设置有助于用户确信：使用 0 将不会损害他们的合法应用程序。另一方面，安全性保护则稍弱一些。

下表汇总了从程序堆栈执行时，使用 **chatr +es** 和 **executable_stack** 的可能组合的结果。运行 **chatr +es disable** 只取决于在确定是否授予堆栈执行权限时的 **executable_stack** 内核可调参数的设置，并等同于不对二进制代码运行 **chatr +es** 。

chatr +es	***executable_stack***	操作
启用 禁用或 **chatr** 未运行	1 1	程序正常运行 程序正常运行
启用 禁用或 **chatr** 未运行	0 0	程序正常运行 程序被终止
启用 禁用或 **chatr** 未运行	2 2	程序正常运行 程序正常运行 同时显示警告

返回值

chatr 成功时返回零。如果命令行内容有语法错误，或者无法执行一个或多个指定文件，则 **chatr** 将返回有关其属性无法修改的文件的信息。如果未指定任何文件，则 **chatr** 将返回十进制的 255。

非法选项

如果使用非法选项，则 **chatr** 将返回出现在第一个非法选项之后的 非选项单词的数量。下面的示例将返回 4：

chatr +b enable +xyz enable +mno enable +pqr enable file

无效参数

如果对一个有效选项使用无效参数，并且未指定文件名，则 **chatr** 将返回 0，如下例所示：

chatr +b <*no argument*>

如果指定一个文件名（无论该文件是否存在），则 **chatr** 将返回已指定文件的数量。下面的示例将返回 3：

chatr <*no argument*> **file1 file2 file3**

无效文件

如果该命令无法处理任何给定的文件，则将返回指定的文件的总数（如果指定了若干选项）。否则，它将返回无法处理的文件的数量。如果 **a2** 没有读/写权限，则下面第一个示例将返回 4，第二个示例将返回 1：

chatr +b enable a1 a2 a3 a4

chatr a1 a2 a3 a4

外部语言环境影响

环境变量

下列国际化变量会影响 **chatr** 的执行：

LANG 确定没有 **LC_ALL** 和其他 **LC_*** 环境变量时本地语言、本地惯例和编码字符集的语言环境类别。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值 **C** （请参阅 *lang*(5) ）而不使用 **LANG** 。

LC_ALL 确定所有语言环境类别的值，它优先于 **LANG** 和其他 **LC_*** 环境变量。

LC_CTYPE 确定字符处理函数的语言环境类别。

LC_MESSAGES 确定应该用来影响写入标准错误的诊断消息的格式和内容的语言环境。

LC_NUMERIC 确定数字格式的语言环境类别。

NLSPATH 确定用于处理 **LC_MESSAGES** 的消息目录的位置。

如果任一国际化变量包含无效设置，则 **chatr** 就会认为所有国际化变量都设置为 **C** 。请参阅 *environ*(5) 。

此外，下列环境变量影响 **chatr** ：

TMPDIR 指定用于保存临时文件的目录（请参阅 *tmpnam*(3S) ）。

举例

将 **a.out** 更改为需要载入型

chatr -q a.out

将使用共享库的程序文件的绑定模式更改为即时和非致命性的。此外，启用 **SHLIB_PATH** 环境变量：

chatr -B immediate -B nonfatal +s enable a.out

禁止共享库 **libfoo.sl** 所依据的共享库 **/usr/lib/libc.sl** 的运行时路径查找：

chatr +l /usr/lib/libc.sl libfoo.sl

假定上次运行 **chatr** 时得出的段索引号为 5，将页面大小更改为 4 KB：

chatr +si 5 +p 4K average64

要设置特定段的修改位，请先查找该段的索引或地址编号。

```
chatr a.out
a.out:
  32-bit ELF executable
  shared library dynamic path search:
    LD_LIBRARY_PATH    enabled  first
    SHLIB_PATH         enabled  second
    embedded path      enabled  third  /CLO/TAHOE_BE/usr/lib/hpux32
  shared library list:
    libsin.so
    libc.so.1
  shared library binding:
    deferred
  global hash table enabled
  global hash table size 100
  shared library mapped private disabled
  shared vtable support disabled
  segments:
    index type     address      flags size
        5 text     04000000     ----c    D (default)
        6 data     40000000     ---m-    L (largest possible)
  executable from stack: D (default)
  kernel assisted branch prediction enabled
  lazy swap allocation for dynamic segments disabled
```

对于采用格式 2 的文本段，请使用以下命令：

chatr +si 5 +m enable a.out

或者

chatr +sa 04000000 +m enable a.out

对于格式 1，请使用以下命令：

chatr +mi enable a.out

警告

这一版本的 **chatr** 命令不再支持下列选项：

- **-n**
- **-q**
- **-M**
- **-N**

- **+getbuckets** *size*
- **+plabel_cache** *flag*
- **+q3p** *flag*
- **+q4p** *flag*

作者

chatr 由 HP 开发。

另请参阅

系统工具

ld(1)	调用链接编辑器
dld.so(5)	动态加载程序

其他信息

a.out(4)	汇编程序、编译程序和链接程序输出
magic(4)	HP-UX 实现方案的幻数
sam(1M)	系统管理员
executable_stack(5)	控制缺省情况下程序堆栈是否可执行

文本和教程

《HP-UX Linker and Libraries Online User Guide》
（请参阅 **+help** 选项）

《HP-UX Linker and Libraries User's Guide》
（有关订购信息，请参阅 *manuals*(5)）

名称

chatr_pa: chatr - 更改程序的内部属性

概要

PA-RISC 32 位 SOM chatr

chatr [**-nqsMN** [**z**|**Z**]] [**-l** *library*] [**-B** *mode*] [**+b** *flag*] [**+dbg** *flag*] [**+es** *flag*] [**+mergeseg** *flag*] [**+gst** *flag*] [**+gstbuckets** *size*] [**+gstsize** *size*] [**+k** *flag*] [**+l** *library*] [**+pd** *size*] [**+pi** *size*] [**+plabel_cache** *flag*] [**+q3p** *flag*] [**+q4p** *flag*] [**+r** *flag*] [**+s** *flag*] [**+z** *flag*] *file* ...

PA-RISC 64 位 ELF chatr

有两种合理的句法格式可用于调用 PA-RISC 64 位 **chatr** 。

FORMAT 1 ： 第一种句法格式与 SOM **chatr** 兼容，可实现向后兼容，并可方便地对仅包含单个文本段和单个数据段的普通文件进行操作：

chatr [**-nqszZ**] [**-l** *library*] [**-B** *mode*] [**+b** *flag*] [**+cd** *flag*] [**+ci** *flag*] [**+es** *flag*] [**+gst** *flag*] [**+gstsize** *size*] [**+k** *flag*] [**+l** *library*] [**+md** *flag*] [**+mi** *flag*] [**+pd** *size*] [**+pi** *size*] [**+s** *flag*] [**+z** *flag*] *file* ...

FORMAT 2 ： 第二种句法格式提供了明确指定要修改的段的功能：

chatr [**-s**] [**-B** *mode*] [**+c** *flag*] [**+dz** *flag*] [**+k** *flag*] [**+m** *flag*] [**+p** *size*] [**+r** *flag*] [**+s** *flag*] [**+si** *index* | **+sa** *address* | **+sall**] [**+z** *flag*] *file* ...

说明

chatr 可用于为 32 位模式 SOM 和 64 位模式 ELF 文件更改程序的内部属性。

完成后，如果没有指定 **-s** ， **chatr** 会将文件的旧值和新值均输出到标准输出。

+pd 和 **+pi** 选项只用于提示虚拟内存页面大小。实际页面大小可能有所不同。在某些情况下， **L** 的页面大小提示可能会提高性能，这取决于应用程序的特定内存要求。

有些应用程序的性能可能会受益于静态分支预测，而其他应用程序则不能。 **+r** 选项提供了使用或避免使用该功能的提示。

+gst 和相关选项使用可提高导出符号搜索功能的全局符号表，从而实现了性能上的改进。有关详细信息，请参阅 *dld.sl*(5) 和《HP-UX Linker and Libraries Online User Guide》。

PA-RISC 32 位 SOM 和 PA-RISC 64 位 ELF（格式 1） chatr 的公共选项

缺省情况下， **chatr** 将每个 *file* 的幻数和文件属性输出到标准输出。

-l *library* 如果提供了目录路径列表，则表示指定的共享库容易受运行时路径查找的影响（请参阅 **+s** 和 **+b**）。

-n 将 *file* 从需要载入型 (**DEMAND_MAGIC**) 更改为共享型 (**SHARE_MAGIC**) （在 PA-RISC 64 位格式 1 中忽略）。

-q 将 *file* 从共享型 (**SHARE_MAGIC**) 更改为需要载入型 (**DEMAND_MAGIC**)（在 PA-RISC 64 位格式 1 中忽略）。

-s 以静默方式执行其操作。（可与 PA-RISC 64 位格式 2 命令一起使用）。

-B *mode* 选择程序（使用共享库）的运行时绑定行为模式。必须指定一种主要绑定模式 **immediate** 或 **deferred** 。也可以指定一个或多个绑定修饰词 **nonfatal** 、 **verbose** 或 **restricted** ，每个修饰词都有一个单独的选项。有关绑定模式的说明，请参阅《HP-UX Linker and Libraries User's Guide》手册。（可与 PA-RISC 64 位格式 2 命令一起使用）。

+b *flag* 控制在构建程序（如果有）时存储的嵌入式路径列表是否可用于查找该程序所需的共享库。两个标志值 **enable** 和 **disable** 分别用于启用和禁用嵌入式路径列表。但不能对 ELF（PA-RISC 64 位）文件使用 **disable** ，否则将发出一条警告消息。请参阅 **+s** 选项。可以使用 **+b** 选项为过滤器库启用嵌入式路径。

+dbg *flag* 以专用方式控制共享库文本段的映射。标志值 **enable** 和 **disable** 用于切换请求的打开和关闭状态。启用之后，可以在专用、可写入区域映射共享库的文本段。在启用 **+mergeseg** 的情况下使用时，可以合并共享库的文本段。

+es *flag* 用 *flag* 值 **enable** 和 **disable** 控制从堆栈执行用户代码的能力。有关安全问题的其他信息，请参阅下文 限制对堆栈的执行权限一节。

+gst *flag* 控制是否将全局符号表散列机制用于查找符号导入（或导出）条目的值。两个标志值 **enable** 和 **disable** 分别用于启用和禁用全局符号表散列机制。缺省值为 **disable** 。

+gstsize *size* 使用全局符号表散列机制请求特定的散列阵列 *size* 。该值介于 1 和 **MAXINT** 之间。缺省值为 1103。将该选项与 **+gst enable** 配合使用。

+k *flag* 请求内核辅助的分支预测。标志 **enable** 和 **disable** 分别用于打开和关闭该请求。（可与 PA-RISC 64 位格式 2 命令一起使用）。

+l *library* 如果提供了目录路径列表，则表示指定的共享库不受运行时路径查找的影响（请参阅 **+s** 和 **+b**）。

+mergeseg *flag* 控制共享库段合并功能。标志值 **enable** 和 **disable** 用于切换该请求的打开和关闭状态。请参阅《HP-UX Linker and Libraries User's Guide》中有关共享库段合并的说明。启用之后，将合并在程序启动时加载的共享库的所有数据段。这样，内核可以使用更大的页面表条目，从而可以提高运行时性能。

+pd *size* 请求应该用于数据的特定虚拟内存页面大小。 **4K** 、 **16K** 、 **64K** 、 **256K** 、 **1M** 、 **4M** 、 **16M** 、 **64M** 、 **256M** 和 **L** 等大小均受支持。 **L** 大小将导致使用可用的最大页面大小。如果无法实现请求的大小，则实际页面大小可能有所不同。

+pi *size* 请求应该用于指令的特定虚拟内存页面大小。有关其他信息，请参阅 **+pd** 选项。

+r *flag* 当执行该程序时请求静态分支预测。标志 **enable** 和 **disable** 分别用于打开和关闭该请求。（可与 PA-RISC 64 位格式 2 命令一起使用）。

+s *flag* 控制用 **SHLIB_PATH** 环境变量指定的目录路径列表是否可用于查找程序所需的共享库。两个标志值 **enable** 和 **disable** 分别用于启用和禁用环境变量。如果同时使用了 **+s** 和 **+b** ，则它们在命

令行上的相对顺序将指示首先搜索哪个路径列表。请参阅 **+b** 选项。（可与 PA-RISC 64 位格式 2 命令一起使用）。

+z 对所有数据段（使用 PA-RISC 32 位 **chatr** 或 PA-RISC 64 位 **chatr** 格式 1）或特定段（使用 PA-RISC 64 位 ELF **chatr** 格式 2）启用惰性交换。不可用于非数据段。

-z,-Z 启用空指针的运行时取消引用来产生 SIGSEGV 信号（这是对 -Z 选项的补充）。

仅适用于 PA-RISC 32 位 SOM chatr 的选项

-M 将 *file* 从 **EXEC_MAGIC** 更改为 **SHMEM_MAGIC** （该选项是一个过渡性解决方案，直至真正的 64 位内核提供了 64 位寻址能力为止。请参阅下面的“chatr 和幻数”以及“使用 SHMEM_MAGIC”）。

-N 将 *file* 从 **SHMEM_MAGIC** 更改为 **EXEC_MAGIC** （该选项是一个过渡性解决方案，直至真正的 64 位内核提供了 64 位寻址能力为止。请参阅下面的“chatr 和幻数”以及注释）。

+gstbuckets *size*

使用全局符号表散列机制请求每个条目的特定存储桶数。该值介于 1 和 **MAXINT** 之间。缺省值为 3。将该选项与 **+gst enable** 配合使用。

+plabel_cache *flag*

控制 plabel 缓存机制的使用。标志 **enable** 和 **disable** 分别用于打开和关闭该请求。缺省值为 **disable** 。将该选项与 **+gst enable** 配合使用。

该选项对 C++ 有效。在 C++ 应用程序中，动态加载程序需要重复访问 PLABEL 信息（导入 stub）。为了加快访问速度，动态加载程序使用全局符号表结构同时来包含 PLABEL 条目。当在 dl_header 结构中设置了 PLABEL_CACHE 标志（启用了 **ld +plabel_cache enable a.out** 或 **chatr +plabel_cache enable a.out** ）时，将启用该行为。

+q3p *flag* 控制标志位设置，以指示 32 位进程如何使用第三象限作为数据空间。

enable 标志设置标志位，以指示 32 位进程使用第三象限作为专用数据空间。通过设置标志位， 32 位进程的专用数据空间可从 1.9GB 增加到 2.85GB。

disable 标志不设置位，这样第三象限会恢复为缺省状态，在这种状态下该象限用作共享内存。

该标志机制与设置第一象限和第二象限用法的机制不同。其通过使用可执行文件的幻数设置这些值。（请参阅 **-M** 和 **-N** 选项）。

+q4p *flag* 控制标志位设置，以指示 32 位进程如何使用第三象限和第四象限作为数据空间。

enable 标志设置标志位，以指示 32 位进程使用第四象限作为专用数据空间。通过设置 **+q4p** 标志位， 32 位进程的专用数据空间可从 1.9GB 增加到 3.8GB。当针对专用数据空间设置第四象限时，第三象限会自动设置以用作专用数据空间，从而忽略当前的 **+q3p** 值。

disable 标志不设置标志位，这样第四象限会恢复为缺省状态，在这种状态下该象限用作共享内存。使用 **+q4p disable** ， **+q3p** 标志的值控制是将第三象限用作专用数据空间还是共享内存。

该标志机制与设置第一象限和第二象限用法的机制不同。其通过使用可执行文件的幻数设置这些值。（请参阅 **-M** 和 **-N** 选项）。

适用于 PA-RISC 64 位 ELF chatr 的选项

PA-RISC 64 位 ELF **chatr** 与 SOM **chatr** 类似，但支持新的选项（同时废弃其他选项）。

新选项：

OPTIONS FOR PA-RISC 64-bit ELF chatr (FORMAT 1)

+cd 为文件的数据段设置代码位。

+ci 为文件的文本段设置代码位。

+md 为文件的数据段设置修改位。

+mi 为文件的文本段设置修改位。

OPTIONS FOR PA-RISC 64-bit ELF chatr (FORMAT 2)

对于公共选项： **-s** 、 **-B** *mode*
、 **+k** *flag* 、 **+r** *flag* 、 **+s** *flag* 、 **+z** *flag* 。

+c 为指定段设置代码位。

+dz 启用或禁用动态分配段（如堆栈或堆）的惰性交换分配。

+m 为指定段设置修改位。

+p 为指定段设置页面大小。

+sa 使用一个地址为一组属性修改指定一个段。

+sall 为一组属性修改使用文件中的所有段。

+si 使用一个段索引号为一组属性修改指定一个段。

chatr 和幻数

术语 共享型应用于幻数 **SHARE_MAGIC** ，同时术语 需要载入型应用于幻数 **DEMAND_MAGIC** 。有关详细信息，请参阅 *magic*(4) 和《HP-UX Linker and Libraries Online User Guide》。

chatr 在输出中标记下列类型的可执行文件。

SHARE_MAGIC：
共享型可执行文件

DEMAND_MAGIC：
需要载入型可执行文件

EXEC_MAGIC：
普通可执行文件

SHMEM_MAGIC：
普通 **SHMEM_MAGIC** 可执行文件

缺省情况下，链接程序生成 **SHARE_MAGIC** 可执行文件。

使用 SHMEM_MAGIC

SHMEM_MAGIC 是一个过渡性解决方案，直至真正的 64 位内核提供了 64 位寻址能力为止。

HP 的 64 位体系结构的未来实现方案（高于 PA-RISC 2.0 的版本），将不支持 **SHMEM_MAGIC** 。在这些体系结构上，那些需要共享内存大于 1.75 GB 的程序必须针对这些体系结构重新编译为 64 位可执行文件。

对于在任何 64 位 HP 实现方案（包括 PA-RISC 2.0）上编译为 64 位可执行文件的程序，不可将其标记为 **SHMEM_MAGIC** ，也没有这个必要，因为这些程序已经能够访问大于 1.75 GB 的共享内存。

通过其他类型的可执行文件得到的额外 1 GB 共享内存，只能用于系统 V 共享内存，而不能用于其他形式的共享内存（如内存映射文件）。

限制对堆栈的执行权限

侵占系统的一种常用方法是，恶意使某程序堆栈发生缓冲区溢出，例如将特别长的、精选命令行参数传递到不需要它们的特权程序。怀有恶意的无权限用户可以利用这种方法欺骗特权程序，使其为自己启动超级用户 Shell，或执行类似的未经授权的操作。

降低此类攻击风险的一个简单而非常有效的方法是，从程序的堆栈页中删除执行权限。这将在不降低系统性能的情况下增强系统安全性，并且对绝大多数的合法应用程序不产生负面影响。本节介绍的更改仅影响极少数的程序，这些程序会试图执行（或被欺骗执行）程序堆栈上的指令。

如果为某个程序启用了本节所述的堆栈保护功能，但该程序试图从堆栈中执行代码，则 HP-UX 内核将终止该程序，同时发出 **SIGKILL** 信号，还显示一条要求查阅本手册页面部分的消息，同时在系统消息日志中记录一条错误消息（使用 **dmesg** 可查看该错误消息）。内核记录的消息为：

WARNING: UID # may have attempted a buffer overflow attack. PID # (program_name) has been terminated. See the '+es enable' option of chatr(1).

如果遇到上述消息之一，请与该程序所有者联系，以便确定该程序是否正在合法地从堆栈中执行代码。如果是，则可以使用下述两种方法或其中一种方法，使程序重新正常运行。如果该程序正在从堆栈中非法地执行代码，则应该猜想是否存在恶意活动并采取适当的操作。

HP-UX 提供了两种允许从程序堆栈合法执行代码的方法。组合使用这两种方法，有助于在安全性和兼容性之间实现站点特定的利弊权衡。

第一种方法是使用 **chatr** 的 **+es** 选项，从而影响单个程序。这种方法通常用于指定某个二进制文件必须能够从其堆栈执行，而无论系统缺省设置如何。这样就允许使用限制性的系统缺省值，同时不禁止合法程序在它们的堆栈上执行代码。从理论上讲，该选项应由程序提供者设置（如果需要），以便将程序安装者进行手动干预的的需要降到最低程度。

另一种方法是设置内核可调参数 **executable_stack** ，从而针对堆栈是否可执行设置一个系统级的缺省值。用 **sam** （请参阅 *sam*(1M)）将 **executable_stack** 参数设置为 1（一），将指示 HP-UX 内核不执行保护程序堆栈。如果与较旧版本的兼容性比安全性更为重要，那么此为首选设置。如果安全性比兼容性更为重要，那么合适的做法是将其设置为 0（零）。此为建议设置，因为它将显著增强系统的安全性，同时对合法应用程序的负面影响（如果有）也最小。

组合使用这些设置能够适合许多应用程序的需要。例如，将 **executable_stack** 设置为 0 后，您可能会发现有一个或两个关键应用程序无法正常工作，这是因为它们需要从堆栈合法地执行。使用自修改代码的模拟程序或解释程序等程序，就属于您可能会遇到的上述程序实例。要获得使用限制性系统缺省值的安全性优点，同时仍然保证这些特定应用程序能够正常运行，请将 **executable_stack** 设置为 0，然后对需要从堆栈执行代码的特定二进制程序运行 **chatr +es enable** 。由于这些二进制代码会输出要求查阅本手册页的错误消息，因此它们在执行时，很容易就能识别出来。

executable_stack 的可能设置如下所示：

executable_stack = 0
（缺省值）设置为 0（缺省值）会导致堆栈无法执行，因此从安全性角度考虑，强烈建议采用此设置。

executable_stack = 1
设置为 1 会导致所有程序堆栈都可以执行，从兼容性角度考虑，此设置是最可靠的，但对于此参数而言，则是安全性最低的设置。

executable_stack = 2
设置为 2 等同于设置为 0，不过该设置会发出非致命性警告，而不是终止试图从其堆栈执行的进程。使用该设置有助于用户确信：使用 0 将不会损害他们的合法应用程序。另一方面，安全性保护则稍弱一些。

下表汇总了从程序堆栈执行时，使用 **chatr +es** 和 **executable_stack** 的可能组合的结果。运行 **chatr +es disable** 只取决于在确定是否授予堆栈执行权限时的 **executable_stack** 内核可调参数的设置，并等同于不对二进制代码运行 **chatr +es** 。

chatr +es	**executable_stack**	操作
启用 禁用或 chatr 未运行	1 1	程序正常运行 程序正常运行
启用 禁用或 chatr 未运行	0 0	程序正常运行 程序被终止
启用 禁用或 chatr 未运行	2 2	程序正常运行 程序正常运行 同时显示警告

返回值

chatr 成功时返回零。如果命令行内容有语法错误，或者无法执行一个或多个指定文件，则 **chatr** 将返回有关其属性无法修改的文件的信息。如果未指定任何文件，则 **chatr** 将返回十进制的 255。

非法选项

对于 PA-RISC 32 位 **chatr** ，如果使用非法选项，则 **chatr** 将返回命令行中的单词的数量。例如，

chatr +b enable +xyz enable 返回 5（因为使用了非法选项 **+xyz** ）。

chatr +b enable +xyz enable +mno file1 file2 返回 8。

对于 PA-RISC 64 位 **chatr**，如果使用非法选项，则 **chatr** 将返回出现在第一个非法选项之后的 非选项单词的数量。

chatr +b enable +xyz enable +mno enable +pqr enable file 返回 4。

无效参数

如果对一个有效选项使用无效参数，并且未指定文件名，则 PA-RISC 32 位和 64 位 **chatr** 都将返回 0。

chatr +b <*no argument*> 返回 0。

对于 PA-RISC 32 位 **chatr** ，如果指定一个文件名（无论该文件是否存在），则 **chatr** 将返回命令行中的单词的数量。

chatr +b <no argument> file 返回 4。

对于 PA-RISC 64 位 **chatr** ，如果指定一个文件名（无论该文件是否存在），则 **chatr** 将返回已指定文件的数量。

chatr +b <*no argument*> **file1 file2 file3** 返回 3。

无效文件

对于 PA-RISC 32 位和 64 位 **chatr** 两者而言，如果该命令无法处理任何给定的文件，则将返回指定的文件的总数（如果指定了若干选项）。否则，它将返回无法处理的文件的数量。

chatr +b enable a1 a2 a3 a4 （其中 **a2** 没有读/写权限）返回 4。

chatr a1 a2 a3 a4 返回 1。

外部语言环境影响

环境变量

下列国际化变量会影响 **chatr** 的执行：

LANG 确定没有 **LC_ALL** 和其他 **LC_*** 环境变量时本地语言、本地惯例和编码字符集的语言环境类别。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值 **C** （请参阅 *lang*(5) ）而不使用 **LANG**。

LC_ALL 确定所有语言环境类别的值，它优先于 **LANG** 和其他 **LC_*** 环境变量。

LC_CTYPE 确定字符处理函数的语言环境类别。

LC_MESSAGES 确定应该用来影响写入标准错误的诊断消息的格式和内容的语言环境。

LC_NUMERIC 确定数字格式的语言环境类别。

NLSPATH 确定用于处理 **LC_MESSAGES** 的消息目录的位置。

如果任一国际化变量包含无效设置，则 **chatr** 就会认为所有国际化变量都设置为 **C** 。请参阅 *environ*(5)。

此外，下列环境变量影响 **chatr** ：

TMPDIR 指定用于保存临时文件的目录（请参阅 *tmpnam*(3S) ）。

举例

将 **a.out** 更改为需要载入型

chatr -q a.out

将使用共享库的程序文件的绑定模式更改为即时和非致命性的。此外，启用 **SHLIB_PATH** 环境变量：

chatr -B immediate -B nonfatal +s enable a.out

禁止共享库 **libfoo.sl** 所依据的共享库 **/usr/lib/libc.sl** 的运行时路径查找：

chatr +l /usr/lib/libc.sl libfoo.sl

假定上次运行 **chatr**
时得出的段索引号为 5，将页面大小更改为 4 KB：

chatr +si 5 +p 4K average64

作者

chatr 由 HP 开发。

另请参阅

系统工具：

ld(1) 调用链接编辑器

其他:

a.out(4)	汇编程序、编译程序和链接程序输出
magic(4)	HP-UX 实现方案的幻数
sam(1M)	系统管理员
executable_stack(5)	控制缺省情况下程序堆栈是否可执行

文本和教程:

《HP-UX Linker and Libraries Online User Guide》
(请参阅 **+help** 选项)

《HP-UX Linker and Libraries User's Guide》
(有关订购信息,请参阅 *manuals*(5))

名称

checknr - 检查 nroff/troff 文件

概要

checknr [**-s**] [**-f**] [**-a.***x1***.***y1***.***x2***.***y2* ... **.***xn***.***yn*] [**-c.***x1***.***x2***.***x3* ...c **.***xn*] [*file* ...]

说明

checknr 搜索 **nroff** 或 **troff** 输入文件，以查找涉及不匹配的开始与结束定界符以及未知命令之类的错误。如果未指定任何文件， **checknr** 会搜索标准输入。 **checknr** 查找下列内容：

- 使用 **\fx** ... **\fP** 进行的字体更改。
- 使用 **\sx** ... **\s0** 进行的大小更改。
- 采取 *open* ... *close* 形式的宏，如 **.TS** 和 **.TE** 宏，必须成对出现。

checknr 了解 **ms** 和 **me** 宏程序包。

选项

checknr 可识别下列选项：

-a 在列表中定义其他宏对。 **-a** 后接多个六字符组，每个组定义一对宏。一个六字符组包含一个句点、第一个宏名称、另一个句点、第二个宏名称。例如，要定义 **.BS** 和 **.ES** 宏对，以及 **.XS** 和 **.XE** 宏对，应使用：

-a.BS.ES.XS.XE

在该选项及其参数之间不允许有空格。

-c 定义 **checknr** 解释为未定义的命令。

-f 忽略 **\f***x* 字体更改。

-s 忽略 **\s***x* 大小更改。

外部语言环境影响

国际代码集支持

支持单字节字符代码集。

诊断信息

checknr 报告定界符不匹配、命令无法识别和命令语法不正确等错误消息。

举例

检查文件 **sorting** 中涉及不匹配的起始与结束定界符以及未知命令的错误，但忽略更改字体引起的错误：

checknr -f sorting

警告

checknr 用于为使用 **checknr** 而准备的文档，非常类似于使用了 **lint** 。它希望 **\f**... 和 **\s**... 命令使用特定的文档书写形式，即每个 **\f***x* 以 **\fP** 终止，每个 **\s***x* 以 **\s0** 终止。尽管直接对后面的字体或点大小进行编码（而不使用 **\fP**

或 **\s0** ）时，文件可以正确格式化，但采用这种方法时 *checknr* 将报告错误。如果要使用 *checknr* 来检查文件，则应使用 **\fP** 和 **\s0** 定界惯例。

-a 不能用于定义单字符宏名称。

checknr 无法识别某些合理的结构，如条件结构。

作者

checknr 由加州大学伯克利分校开发。

另请参阅

checkeq(1)、lint(1)、nroff(1)。

名称

chfn - 更改用户信息；由 finger 使用

概要

chfn [*login-name*]

chfn -r files [*login-name*]

chfn -r nis [*login-name*]

chfn -r nisplus [*login-name*]

chfn -r dce [*login-name*]

说明

chfn 命令可更改在 **repository** 中为当前登录的用户或由 *login-name* 指定的用户所存储的用户信息（请参阅 *passwd*(1)）。

在口令文件条目的保留字段（第 5 个）内，此信息被组织为由逗号分隔的四个子字段形式。该字段依次由用户的全名、位置代码、办公室电话号码和家庭电话号码组成。此信息由 **finger** 命令和其他程序使用（请参阅 *finger*(1)）。

chfn 提示您输入每个子字段。提示中包括缺省值（括在方括号中）。可按 Return 键接受缺省值。要输入空白子字段，请键入 **none** 一词。

DCE 储备库 (**-r dce**) 只有在配置了集成登录后才可用，请参阅 *auth.adm*(1M)。如果已配置了集成登录，则还要注意其他事项。具有适当 DCE 权限的用户可以修改用户的 finger (gecos) 信息；该操作并不要求具有超级用户权限。

如果未指定储备库，即 **chfn** [*login-name*]，则将只在 passwd 文件中更改 finger 信息。

在运行 **chfn** 后运行 **finger** 可以确保信息得到正确处理。

选项

可以识别下列选项：

-r 指定要应用该操作的储备库。支持的储备库包括 **files**、**nis**、**nisplus** 和 **dce**。

子字段值

Name 最多为 1022 个可打印字符。

finger 命令和其他实用程序将扩展在此子字段的任意位置找到的 **&**，方法是用登录名替换它并将登录名的首字母更改为大写（chfn 不更改输入 **&**）

Location 最多为 1022 个可打印字符。

Office Phone 最多为 25 个可打印字符。

如果该值仅包含数字，则 **finger** 会插入适当的连字符。

Home Phone 最多为 25 个可打印字符。

如果该值仅包含数字，则 **finger** 会插入适当的连字符。

安全限制

您必须拥有适当的权限，才能使用可选的 *login-name* 参数来更改另一个用户的信息。

举例

下面是一个运行示例。用户的输入以常规类型显示。

Name [Tracy Simmons]:
Location (Ex: 47U-P5) []: 42L-P1
Office Phone (Ex: 1632) [77777]: 71863
Home Phone (Ex: 9875432) [4085551546]: none

警告

办公室和扩展信息的编码与安装相关。

由于历史原因，用户名等信息存储在 **/etc/passwd** 文件中。此位置不适合于存储这些信息。

由于两个用户可能会尝试同时向 **passwd** 文件中写入数据，因此开发了一种同步方法。在个别情况下， **chfn** 会输出一条消息，指出口令文件正忙。如果出现此情况， **chfn** 会“休眠”较短的一段时间，然后再次尝试向 **passwd** 文件写入数据。

作者

chfn 由加州大学伯克利分校开发。

文件

/etc/passwd
/etc/ptmp

注释

chfn 命令是指向 **passwd** 命令的硬链接。当执行 **chfn** 时，实际上会使用相应的参数执行 **passwd** 命令，以更改在命令行中指定的 *repository* 中的用户 gecos 信息。如果未指定任何 *repository* ，则会在 **/etc/passwd** 文件中更改 gecos 信息。

另请参阅

chsh(1)、 finger(1)、 passwd(1)、 passwd(4)。

名称

chkey - 更改用户的安全 RPC 密钥对

概要

chkey [**-p**] [**-s nisplus | nis | files**]

说明

chkey 用于更改用户的安全 RPC 公用密钥和私用密钥对。 **chkey** 会提示您输入旧的安全 RPC 口令，并通过解密私用密钥验证该口令是否正确。如果用户还未曾注册密钥，那么 **chkey** 会使用本地 *keyserv*(1M) 守护程序注册该私用密钥。如果安全 RPC 口令与登录口令不匹配， **chkey** 会提示您输入登录口令。 **chkey** 使用登录口令来加密用户的私用 Diffie-Hellman（192 位）加密密钥。

chkey 可确保登录口令与安全 RPC 口令保持相同。

密钥对可以存储在 **/etc/publickey** 文件（请参阅 *publickey*(4) ）、NIS **publickey** 映射或 NIS+ **cred.org_dir** 表中。如果生成了新的私用密钥，则本地 *keyserv*(1M) 守护程序将注册该密钥。

如果未使用 **-s** 选项指定 **publickey** 的来源，则 **chkey** 会参考名称服务交换配置文件（参阅 *nsswitch.conf*(4) ）中的 **publickey** 条目。如果 **publickey** 条目指定且只指定了一个来源，那么 **chkey** 将更改指定名称服务中的密钥。但是，如果列出了多个名称服务，则 **chkey** 无法确定要更新哪个来源，并将显示错误消息。用户应使用 **-s** 选项明确地指定来源。

不允许非超级用户在 **/etc/publickey** 文件中更改他们的密钥对。

选项

-p 使用用户的登录口令重新加密现有私用密钥。

-s nisplus 更新 NIS+ 数据库。

-s nis 更新 NIS 数据库。

-s files 更新 **files** 数据库。

作者

chkey 由 Sun Microsystems, Inc. 开发。

文件

/etc/nsswitch.conf

/etc/publickey

另请参阅

keylogin(1)、 keylogout(1)、 keyserv(1M)、 newkey(1M)、 nisaddcred(1M)、 nsswitch.conf(4)、 publickey(4)。

名称

chmod - 更改文件模式访问权限

概要

/usr/bin/chmod [**-A**] [**-R**] *symbolic_mode_list file* ...

过时形式:

/usr/bin/chmod [**-A**] [**-R**] *numeric_mode file* ...

说明

chmod 命令根据 *symbolic_mode_list* 或 *numeric_mode* 的值更改一个或多个 *file* 的权限。可以使用 **ls -l** 命令显示文件的当前权限（请参阅 *ls*(1) ）。

符号模式列表

symbolic_mode_list 是以下格式的逗号分隔操作列表。不允许有空白字符。

[*who*]*op*[*permission*] [**,**...]

变量字段可以具有下列值:

who 下列一个或多个字母:

- **u** 修改用户（所有者）的权限。
- **g** 修改组的权限。
- **o** 修改其他用户的权限。
- **a** 修改所有用户的权限（ a 等效于 **ugo** ）。

op 必需项；下列符号之一:

- **+** 将 *permission* 添加到 *who* 的现有文件模式位。
- **-** 从 *who* 的现有文件模式位中删除 *permission* 。
- **=** 将 *who* 的现有模式位替换为 *permission* 。

permission 下列一个或多个字母:

- **r** 添加或删除 *who* 的读取权限。
- **w** 添加或删除 *who* 的写入权限。
- **x** 添加或删除 *who* 的执行文件（搜索目录）权限。
- **s** 添加或删除 *who* 的“在文件执行时设置所有者 ID”或“在文件执行时设置组 ID”权限。仅当在 *who* 中表示或隐含 **u** 或 **g** 时，该选项才有用。
- **t** 添加或删除“在文件执行时保存文本图像”（粘着位）权限。仅当在 *who* 中表示或隐含 **u** 时，该选项才有用。请参阅 *chmod*(2) 。
- **X** 有条件地添加或删除执行（或搜索）权限，如下所示:
  - 如果 *file* 是一个目录，则将搜索权限添加到 *who* 的现有文件模式或删除该搜索权限（与 **x** 相同）。

- 如果 *file* 不是一个目录，并且当前文件权限包括至少一个用户、组或其他用户的执行权限（ ls **-l** 显示 **x** 或 **s** ），则添加或删除 *who* 的执行文件权限。
- 如果 *file* 不是一个目录，且在当前文件模式下未设置执行权限，则不更改任何执行权限。

或者仅下列字母之一：

u 将当前用户权限复制到 *who* 。
g 将当前组权限复制到 *who* 。
o 将当前的其他用户权限复制到 *who* 。

操作是按指定顺序执行的，并且可以覆盖在同一命令行中指定的以前的操作。

如果省略 *who* ，则更改所有用户的 **r** 、 **w** 、 **x** 和 **X** 权限，前提是当前文件模式创建掩码允许这些更改（请参阅 *umask*(1) ）。更改 **s** 和 **t** 权限，好像在 *who* 中指定了 **a** 。

仅当与 **=** 一起使用以删除所有权限时，省略 *permission* 才是有用的。

数字模式（已过时）

通过指定 *numeric_mode* （从下列模式位的逻辑“或”（和）构建的八进制数），可以设置绝对权限：

其他模式位：

4000 (= **u=s**) Set user ID on file execution (file only)
2000 (= **g=s**) Set group ID on file execution (file only)
1000 (= **u=t**) Set sticky bit; see below and *chmod*(2)

权限模式位：

0400 (= **u=r**) Read by owner
0200 (= **u=w**) Write by owner
0100 (= **u=x**) Execute (search in directory) by owner
0040 (= **g=r**) Read by group
0020 (= **g=w**) Write by group
0010 (= **g=x**) Execute/search by group
0004 (= **o=r**) Read by others
0002 (= **o=w**) Write by others
0001 (= **o=x**) Execute/search by others

选项

-A 保留与文件关联的任何可选访问控制列表 (ACL) 条目（仅适用于 HFS 文件系统）。缺省情况下，为了符合 IEEE 标准 POSIX 1003.1-1988，将删除可选的 HFS ACL 条目。对于 JFS ACL，该选项不起作用，因为始终保留可选的 JFS ACL 条目。有关访问控制列表的信息，请参阅 *acl*(5) 和 *aclv*(5) 。

-R 以递归方式更改文件模式位。对于指定目录的每个 *file* 操作数， **chmod** 将更改指定目录以及文件层次结构中它下面的所有文件和子目录的文件模式位。

只有文件的所有者或具有相应特权的用户才能更改其模式。

只有具有相应特权的用户才能设置（如果以前已设置，则为保留）常规文件的粘着位。

如果在目录上设置了粘着位，则该目录内的文件只能由文件的所有者、目录的所有者或超级用户重命名或删除（即使目录的模式另行允许这样的操作时，也不例外）。

为了设置“设置组 ID”位，文件的组必须对应于当前的组 ID。

如果 **chmod** 用于符号链接，则更改该链接所引用的文件模式。

外部语言环境影响

环境变量

LC_MESSAGES 用于确定显示消息的语言。

如果未指定 **LC_MESSAGES** 或它为空，则它缺省为 **LANG** 的值。如果未指定 **LANG** 或它为空，则它缺省为 **C** （请参阅 *lang*(5) ）。

如果任一国际化变量中包含无效设置，则所有国际化变量都会缺省为 **C** 。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

完成后， **chmod** 将返回下列值之一：

0 成功完成。

>0 出错情形。

举例

拒绝将写入权限授予其他用户：

chmod o-w *file*

使某个文件可以由每个用户执行：

chmod a+x *file*

将读取和执行权限分配给每个用户，并设置“设置用户 ID”位：

chmod a=rx,u+s *file*

将读取和写入权限分配给文件所有者，并将读取权限分配给其他每个用户：

chmod u=rw,go=r *file*

或过时形式：

chmod 644 *file*

遍历目录子树，使所有常规文件只能由用户和组读取，并使每个用户都可以执行（搜索）所有可执行文件和目录：

chmod -R ug+r,o-r,a+X *pathname*

如果 **umask** 的当前值为 **020** （umask **-S** 显示 **u=rwx,g=rx,o=rwx**；不更改组的写入权限），并且文件 **mytest** 的当前权限为 **444** (**a=r**)，由 **ls -l** 显示为 **-r--r--r--**，则命令

chmod +w mytest

将权限设置为 **646** (**uo=rw,g=r**)，由 **ls -l** 显示为 **-rw-r--rw-**。

如果 **umask** 的当前值为 **020** （umask **-S** 显示 **u=rwx,g=rx,o=rwx**；不更改组的写入权限），并且文件 **mytest** 的当前权限为 **666** (**a=rw**)，由 **ls -l** 显示为 **-rw-rw-rw-**，则命令

chmod -w mytest

将权限设置为 **464** (**uo=r,g=rw**)，由 **ls -l** 显示为 **-r--rw-r--**。

相关内容

-A 选项会导致 **chmod** 在不支持 ACL 的文件系统上失败。

作者

chmod 由 AT&T 和 HP 开发。

另请参阅

chacl(1)、 ls(1)、 umask(1)、 chmod(2)、 acl(5)、 aclv(5)。

符合的标准

chmod：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

chown、chgrp - 更改文件所有者或组

概要

chown [**-h**] [**-R**] *owner*[**:***group*] *file* ...

chgrp [**-h**] [**-R**] *group file* ...

说明

chown 命令将每个指定 *file* 的所有者 ID 更改为 *owner* ，并将每个指定 *file* 的组 ID 更改为 *group* （可选）。

chgrp 命令将每个指定 *file* 的组 ID 更改为 *group* 。

owner 可以是十进制用户 ID 或 **/etc/passwd** 文件中的登录名。

group 可以是十进制组 ID 或 **/etc/group** 文件中的组名。

要更改所有者或组，必须拥有该文件并拥有 CHOWN 特权（请参阅 *setprivgrp*(1M) ）。如果超级用户之外的用户对常规文件调用任一命令，则将清除文件模式的设置用户 ID 位和设置组 ID 位（分别是 04000 和 02000）。请注意，使用 **setprivgrp** ，可以限制给定用户或组使用该命令的能力（请参阅 *setprivgrp*(1M) ）。

访问控制列表 - 仅限 HFS 文件系统

通过在文件的访问控制列表中设置可选的 ACL 条目，用户可以允许或拒绝个人或组访问该文件（请参阅 *acl*(5)）。当 **chown** 与 HFS ACL 一起使用时，如果文件的新所有者和（或）组在文件访问控制列表中不具有对应于 *user***.%** 和（或） **%.***group* 的可选 ACL 条目，则该文件的访问权限位将保持不变。但是，如果已经在文件的 ACL 中使用 *user***.%** 和（或） **%.***group* 的可选 ACL 条目指定新的所有者和（或）组， **chown** 会将相应的文件访问权限位（和相应的基 ACL 条目）设置为该条目中包含的权限。

访问控制列表 - 仅限 JFS 文件系统

通过在文件的访问控制列表中设置可选的 ACL 条目，用户可以允许或拒绝个人或组访问该文件（请参阅 *aclv*(5)）。当 **chown** 与 JFS ACL 一起使用时，如果文件的新所有者和（或组）在文件的访问控制列表中拥有对应于 **user:***uid***:***perm* 和（或） **group:***gid***:***perm* 的可选 ACL 条目，这些条目将留在 ACL 中，但不再起作用，它们被文件的 **user::***perm* 和（或） **group::***perm* 条目所替代。

选项

chown 和 **chgrp** 可识别下列选项:

-h 更改符号链接的所有者或组。

缺省情况下，将更改符号链接指向的目标文件的所有者或组。使用 **-h** ，将不影响符号链接指向的目标文件。如果目标文件是目录，并且您指定 **-h** 和 **-R** ，则不会发生递归操作。

-R 以递归方式更改所有者或组。对于每个将目录命名的 *file* 操作数，将更改该目录以及其下文件层次结构中所有文件和子目录的所有者或组。

外部语言环境影响

环境变量

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_MESSAGES** 或将其设置为空字符串，则会将 **LANG** 的值用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5)）而非 **LANG** 。

如果任一国际化变量中包含无效设置，则 **chown** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

chown 和 **chgrp** 返回下列值之一：

0 成功完成。

>0 发生了错误。

举例

以下命令将文件 **jokes** 的所有者更改为 **sandi** ：

chown sandi jokes

以下命令搜索目录 **design_notes** 并将该目录中的每个文件更改为所有者 **mark** 和组 **users** ：

chown -R mark:users design_notes

警告

从 HP-UX r10.0 开始，已经更改了符号链接的 **chown** 和 **chgrp** 的缺省操作。使用 **-h** 选项可获取以前的缺省操作。

文件

/etc/group

/etc/passwd

另请参阅

chmod(1)、 setprivgrp(1M)、 chown(2)、 group(4)、 passwd(4)、 acl(5)、 aclv(5)。

符合的标准

chown：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

chgrp：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

chsh - 更改缺省登录 Shell

概要

chsh *login-name* [*shell*]

chsh -r files *login-name* [*shell*]

chsh -r nisplus *login-name* [*shell*]

chsh -r nis *login-name* [*shell*]

chsh -r dce *login-name* [*shell*]

说明

chsh 命令可以更改数据库中用户的登录名的登录 Shell （请参阅 *passwd*(1) ）。

DCE 数据库 (**-r dce**) 只有在配置了集成登录 (Integrated Login) 后才可用，请参阅 *auth.adm*(1M) 。如果已经配置了集成登录，则适用其他注意事项。具有适当 DCE 权限的用户可以修改用户的 Shell；该操作并不要求具有超级用户权限。

如果未指定数据库，即 **chsh** [*login-name*] ，则将只在 passwd 文件中更改登录 Shell。

先运行 **chsh** 再运行 **finger** 可确保信息的正确处理。

参数

login-name 用户的登录名。

Shell Shell 的绝对路径名。如果 **/etc/shells** 文件存在，则新的登录 Shell 必须在该文件中列出。否则，您可以指定 *getusershell*(3C) 手册条目中列出的标准 Shell 之一。如果省略了 *Shell* ，则缺省为 POSIX Shell **/usr/bin/sh** 。

选项

可以识别以下选项：

-r 指定要应用该操作的数据库。支持的数据库包括 **files** 、 **nis** 、 **nisplus** 和 **dce** 。

安全限制

您必须拥有适当的权限才能使用可选的 *login-name* 参数来更改其他用户的登录 Shell。

网络功能

NFS

文件 **/etc/passwd** 可以作为一个网络信息服务 (NIS) 数据库实现。

举例

若要将用户 **voltaire** 的登录 Shell 更改为缺省值：

chsh voltaire

若要将用户 **descartes** 的登录 Shell 更改为 C Shell：

chsh descartes /usr/bin/csh

若要将用户 **aristotle** 的登录 Shell 更改为 DCE 注册表中的 Korn Shell：

chsh -r dce aristotle /usr/bin/ksh

警告

由于可能有很多用户试图同时写入 **/etc/passwd** 文件，为此提供了一种口令锁定机制。如果在随后的重新尝试后锁定失败，则 **chsh** 将会终止。

作者

chsh 由 HP 和加州大学伯克利分校联合开发。

注意

chsh 命令是一个指向 **passwd** 命令的硬链接。实际执行 **chsh** 时，将执行具有适当参数的 **passwd** 命令，以更改命令行中指定的 *repository* 中的用户登录 Shell。如果未指定任何 *repository* ，则会在 **/etc/passwd** 文件中更改登录 Shell。

文件

/etc/shells
/etc/ptmp

另请参阅

chfn(1)、 csh(1)、 ksh(1)、 passwd(1)、 sh(1)、 sh-posix(1)、 getusershell(3C)、 pam(3)、 passwd(4)、 shells(4)。

名称

ci - 签入 RCS 修订

概要

ci [*options*] *file*...

说明

ci 可将新修订保存到 RCS 文件中。每个文件名以 **,v** 结尾的文件将被视为 RCS 文件，所有其他文件都假定为工作文件。 **ci** 将每个工作文件的内容保存到相应的 RCS 文件中（请参阅 *rcsintro*(5) ）。

如果 RCS 文件不存在，则 **ci** 会创建一个 RCS 文件，并将工作文件的内容作为初始版本保存在其中。缺省版本号为“1.1”。访问列表初始化为空。 **ci** 将请求说明性文本而不是日志消息（请参阅下面的 **-t** 选项）。

由 **ci** 创建的 RCS 文件会继承工作文件的读取和执行权限。如果 RCS 文件存在，则 **ci** 会保留其读取和执行权限。 **ci** 总是关闭 RCS 文件的所有写入权限。

命令的调用者必须拥有对包含 RCS 文件和工作文件的目录的读/写权限，还必须拥有对 RCS 文件本身的读取权限。在此过程中会创建许多临时文件。在包含 RCS 文件的目录中将创建一个信号量文件。 **ci** 总是创建一个新的 RCS 文件，并取消与旧文件的链接；因此指向 RCS 文件的链接毫无用处。

要使 **ci** 正常工作，用户的登录信息必须位于访问列表中，除非访问列表是空的。用户是文件的所有者或是超级用户。

通常情况下， **ci** 会检查要保存的修订版是否不同于前一版本。如果二者相同， **ci** 将中止保存操作（如果给定了 **-q**），或者询问用户是否要中止（如果省略了 **-q** ）。使用 **-f** 选项可以强制执行保存操作。

如果内存不足，无法检查要保存的修订版与前一版本之间的差异，则可以增加 **swap** 或 **maxdsiz** 的值。

对于所保存的每一修订版， **ci** 都会给出日志消息。日志消息应该概括了所做的更改，它必须使用包含单个“.”或 Control-D 的命令行终止。如果正在签入几个文件，则 **ci** 会询问是否重用前一文件的日志消息。如果标准输入不是终端，则 **ci** 不会给出提示，并为所有文件使用相同的日志消息（请参阅下面的 **-m** 选项）。

所保存修订的版本号可以使用以下任一选项给出： **-r** 、 **-f** 、 **-k** 、 **-l** 、 **-u** 或 **-q** （请参阅下面的 **-r** 选项）。

要为现有分支添加新的修订版，该分支的主修订版必须由调用者锁定。否则，只能创建一个新分支。这一限制对于文件所有者而言不是强制性的，除非将锁定设置为 **strict** （请参阅 *rcs*(1) ）。其他人设置的锁定可通过 **rcs** 命令解除（请参阅 *rcs*(1) ）。

选项

-f[*rev*] 强制进行保存。将保存新修订版，即使新修订版与前一版本相同也是如此。

-k[*rev*] 在工作文件中搜索关键字值，以确定其版本号、创建日期、作者和状态（请参阅 *co*(1) ），并将这些值赋予保存的修订版，而不是在本地进行计算。使用命令选项给出的版本号将覆盖工作文件中的版本号。该选项对于软件分发十分有用。对于发送到多个站点的修订版，应该在这些站点使用 **-k** 选项签入，从而保留其原始版本号、日期、作者和状态。

-l[*rev*] 工作方式类似于 **-r** ，只不过它对保存的修订版执行附加的 **co -l** 。因此，保存的修订版会立即重新签出并被锁定。在保存修订版、而有人希望在签入后继续编辑该版本时，此选项非常有用。

-m"*msg***"** 将 *msg* 字符串用作所有已签入修订版的日志消息。

-n"*name***"** 将符号名称 *name* 指定给已签入的修订版。如果 *name* 已指定给其他版本号，则 **ci** 将输出错误消息。

-N"*name***"** 与 **-n** 相同，只不过它将覆盖前一个 *name* 指定。

-q[*rev*] 静态模式；不显示诊断输出。除非给定了 **-f** 选项，否则，不保存与前一版本相同的修订版。

-r[*rev*] 将 *rev* 版本号指定给已签入的修订版，解除相应的锁定，并删除工作文件。这是缺省值。

如果省略了 *rev* ，则 **ci** 会从调用者的最后一个锁定中派生新的版本号。如果调用者已经锁定了一个分支的主修订版，则新修订版会添加到该分支的主干上，并为新修订版指定一个新版本号。新版本号是通过对主修订版的版本号做增量增加而得到的。如果调用者锁定了非主修订版，则锁定的修订版上会启动一个新分支，锁定修订版的版本号将进行增量增加。缺省的初始分支和级别号都是 1。如果调用者未进行任何锁定，但调用者是文件的所有者，且锁定未设置为 *strict* ，则修订版将添加到主干的主要部分中。

如果 *rev* 指明了版本号，则它必须高于 *rev* 所属分支中最新的版本号；否则，必须启动一个新分支。

如果 *rev* 指明了一个分支而不是修订版，则新修订版将添加到分支的主要部分中。级别数是通过对该分支的主修订版版本号做增量增加而得到的。如果 *rev* 指明了一个并不存在的分支，则将使用初始修订版版本号 *rev***.1** 创建该分支。

注释：在主干中，可以将修订版添加到主要部分中，但不能插入。

-s"*state***"** 将已签入的修订版的状态设置为标识符 *state* 。缺省值为 **Exp** 。

-t[*txtfile*] 将说明性文本写入 RCS 文件中（删除现有文本）。如果省略了 *txtfile* ，则 **ci** 将提示用户输入从标准输入中得到的文本，该文本是使用包含单个 **.** 或 Ctrl-D 的命令行终止的。否则，将从 *txtfile* 文件中复制说明性文本。在初始化过程中，即使未给定 **-t** ，也会请求说明性文本。如果标准输入不是终端，则不会显示提示。

-u[*rev*] 与 **-l** 类似，例外的是所保存的修订版未锁定。如果有人希望在签入修订版后立即对它进行处理（例如编译），则此选项非常有用。

访问控制列表 (ACL)

不应将可选的 ACL 条目添加到 RCS 文件中，因为这可能会导致它们被删除。

诊断信息

对于每个修订版， **ci** 都可以输出 RCS 文件、工作文件，以及已保存修订版的版本号和前一版本的版本号。退出状态总是与最后一个签入的文件有关。如果操作成功，其值为 0；如果操作不成功，其值为 1。

举例

如果当前目录包含子目录 **RCS** ，以及 RCS 文件 **io.c,v** ，则以下所有命令都可以将 **io.c** 的最新修订版保存到 **RCS/io.c,v** 中：

```
ci io.c
ci RCS/io.c,v
ci io.c,v
ci io.c RCS/io.c,v
ci io.c io.c,v
ci RCS/io.c,v io.c
ci io.c,v io.c
```

通过 **Bug fix** 消息，签入 RCS 文件 **foo.c,v** 的 1.2 版：

```
ci -r1.2 -m"Bug Fix" foo.c,v
```

警告

RCS 文件的名称是通过将 **,v** 附加到工作文件名称的末尾而生成的。如果得到的 RCS 文件名称对于 RCS 文件应驻留的文件系统而言太长，则 **ci** 将终止并生成错误消息。

日志消息不能超过 2046 字节。

如果一个文件大约有 240 个修订版，则可能会导致散列表溢出。 **ci** 无法向该文件添加其他修订版，直至删除了一些旧修订版为止。使用 **rcs -o** （o 表示过时版本）命令选项，可以删除旧修订版。

RCS 设计为仅用于文本文件。如果试图对非文本（二进制）文件使用 RCS，则会导致数据损坏。

作者

ci 由 Walter F. Tichy 开发。

另请参阅

co(1)、 ident(1)、 rcs(1)、 rcsdiff(1)、 rcsmerge(1)、 rlog(1)、 rcsfile(4)、 acl(5)、 rcsintro(5)。

名称

ckconfig - 验证所有 FTP 配置文件的路径名。

概要

/usr/bin/ckconfig [**-V**]

说明

ckconfig 实用程序用于验证 FTP 配置文件的下列路径名： **/etc/ftpd/ftpusers** 、 **/etc/ftpd/ftpaccess** 、 **/etc/ftpd/ftpconversions** 、 **/etc/ftpd/ftpgroups** 、 **/etc/ftpd/ftphosts** 、 **/var/adm/syslog/xferlog** 和 **/etc/ftpd/pids/*** 。

该实用程序将检查是否所有 FTP 配置文件都位于指定的路径中。如果在指定的路径中没有找到配置文件，它将向系统管理员发出一条错误消息。

-V 选项将使程序显示版权和版本信息，然后终止。

文件

/usr/bin/ckconfig

作者

ckconfig 由密苏里州华盛顿大学圣路易斯分校开发。

另请参阅

ftpusers(4)、 ftpconversions(4)、 ftpaccess(4)、 ftphosts(4)、 ftpgroups(4)、 xferlog(5)。

名称

cksum - 输出文件校验和与文件大小

概要

cksum [*file* ...]

说明

cksum 命令计算每个命名文件的校验和与该文件中的八位字节数，并将这些信息和文件名输出到标准输出。

cksum 使用基于 32 位循环冗余检验的可移植算法。比起 **sum** 使用的 16 位算法，这种算法可以找到更多的错误（请参阅 *sum*(1)）。CRC 是下列表达式的和，其中 *x* 是文件的每个字节。

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + x^{0}$$

计算的结果被截断为 32 位值。同时输出文件中字节的数目。

如果未指定文件名，则使用标准输入。

cksum 通常用于在系统之间复制文件时验证数据完整性。

外部语言环境影响

环境变量

LANG 用于确定在 **LC_ALL** 和相应的环境变量（以 **LC_** 开头）都未指定语言环境时，语言环境类别将使用的语言环境。如果未设置 **LANG** 或将其设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5)）。

LC_CTYPE 用于确定将语言环境文本字节序列解释为何种字符（例如参数和输入文件中单字节字符和多字节字符）。

LC_MESSAGES 用于确定显示消息的语言。

如果任一国际化变量包含无效设置，则 **cksum** 将认为所有国际化变量都设置为“C”。请参阅 *environ*(5)。

返回值

完成后， **cksum** 将返回下列值之一：

- **0** 所有文件被成功处理。
- **>0** 无法读取一个或多个文件或发生其他错误。

如果遇到不可访问的文件， **cksum** 会继续处理所有剩余文件，但最终退出状态会受到影响。

另请参阅

sum(1)、 wc(1)。

符合的标准

cksum：XPG4、POSIX.2

名称

clear - 清除终端屏幕

概要

clear

说明

clear 如果可能，清除终端屏幕。它会读取终端类型的 **TERM** 环境变量，然后读取相应的 **terminfo** 数据库以确定清除屏幕的方式。

文件

/usr/share/lib/terminfo/?/* 终端数据库文件

作者

clear 由加州大学伯克利分校开发。

另请参阅

terminfo(4)。

名称

cmp - 比较两个文件

概要

cmp [**-l**] [**-s**] *file1 file2* [*skip1* [*skip2*]]

说明

cmp 比较两个文件（如果将 *file1* 或 *file2* 指定为 **-** ，则使用标准输入）。在缺省选项下，如果两个文件相同则 **cmp** 不输出任何消息；如果两个文件不同，则声明出现差别的字节号及行号。如果其中一个文件是另一个文件的起始部分，则该命令会声明这种情况。 *skip1* 和 *skip2* 分别是 *file1* 和 *file2* 的初始字节偏移；可以是八进制或十进制数；数字的形式由环境变量 **LC_NUMERIC** 决定（C 语言中，以 0 开头的数字为八进制数）。请参阅 *environ*(5) 和 *strtol*(3C) 的 **LANG** 。

cmp 可识别下列选项：

-l 输出找到的每个区别所在的字节号（十进制）和有区别的字节（八进制），字节从 1 而不是 0 开始进行编号。

-s 对于有区别的文件不输出任何消息；只返回代码。

外部语言环境影响

环境变量

LANG 用于确定显示消息的语言。如果不指定 **LANG** 或将其设为空字符串，则使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量中包含无效设置，则 **cmp** 就会认为所有国际化变量都设置为“C”。请参阅 *environ5* 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

cmp 返回下列退出值：

0 文件完全相同。
1 文件不完全相同。
2 参数无法访问或缺失。

如果比较直到 file1(file2) 的结尾才成功，则 **cmp** 会输出如下警告。

cmp: EOF on file1(file2)

另请参阅

comm(1)、 diff(1)。

符合的标准

cmp： SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

co - 签出 RCS 修订

概要

co [*options*] *file* ...

说明

co 从 RCS 文件中检索修订。文件名以 **,v** 结尾的每个文件将被视为 RCS 文件。假定所有其他文件为工作文件。 **co** 从每个 RCS 文件中检索修订，并将其存储在相应的工作文件中（另请参阅 *rcsintro*(5) ）。

RCS 文件的修订可以签出为锁定状态或未锁定状态。锁定修订可以防止重叠更新。签出供读取或处理（例如编译）的修订需要设为锁定。通常情况下，签出供编辑并随后签入的修订必须设为锁定。如果某个修订当前被其他用户锁定，则再对其进行锁定会导致失败。使用 **rcs** 命令可以解除锁定，但当同时进行独立的更改时会引发固有的风险（请参阅 *rcs*(1) ）。如果进行锁定，则 **co** 要求调用者位于 RCS 文件的访问列表中，除非：他是文件的所有者、具有适当权限的用户，或访问列表为空。如果不进行锁定，则 **co** 不受访问列表限制的影响。

通过编号、签入日期/时间、作者或状态，可以选择修订。如果这些选项均未指定，则将检索主干上的最新修订。如果这些选项组合应用，则将检索满足所有选项的最新修订。日期/时间、作者和状态的选项在所选分支上检索修订。所选分支是由修订编号（如果给出）派生而来，或是主干上最高的分支。修订编号可以附加到 **-l** 、 **-p** 、 **-q** 或 **-r** 选项。

命令的调用者必须对工作目录有写入权限、对 RCS 文件有读取权限、对包含 RCS 文件的目录有读取权限（用于读取）或读/写权限（用于锁定）。

工作文件从 RCS 文件继承读取和执行权限。此外，除非文件签出为未锁定状态，并且锁定设置为 **strict** （请参阅 *rcs*(1) ），否则将打开所有者的写入权限。

如果具有工作文件名的文件已存在，并且具有写入权限，则 **co** 会在给出 **-q** 时中止签出，或在未给出 **-q** 时询问是否要中止。如果现有工作文件不可写，它将在签出前被删除。

在此过程中会创建许多临时文件。在 RCS 文件所在的目录中会创建一个信号量文件，以防止同时更新。

应用于没有修订的 RCS 文件的 **co** 命令，将创建零长度文件。 **co** 始终执行关键字替换（请参阅下文）。

选项

-l[*rev*]　为调用者锁定签出的修订。如果忽略该选项，则签出的修订不被锁定。有关处理修订编号 *rev* 的信息，请参阅 **-r** 选项。

-p[*rev*]　在标准输出中输出检索到的修订，而不是将其存储在工作文件中。当 **co** 是管道的一部分时，该选项十分有用。

-q[*rev*]　静态模式；不输出诊断信息。

-d*date*　在所选分支上检索签入日期/时间小于或等于 *date* 的最新修订。日期和时间可以采用任意格式给出，然后转换为当地时间。 *date* 格式的示例如下：

Tue-PDT, 1981, 4pm Jul 21	（任意格式）
Fri April 16 15:52:25 EST 1982	（*ctime*(3C) 输出）
4/21/86 10:30am	（格式：*mm***/***dd***/***yy hh***:***mm***:***ss*）

日期和时间中的大多数字段可以是缺省值。 **co** 按年、月、日、小时、分钟和秒的顺序（即从重要性最大到最小的顺序）确定缺省值。必须至少提供其中一个字段。如果省略字段的重要性要高于所提供的最重要字段，则假定为当前值。对于所有其他省略字段，假定为最低的可能值。例如，日期 **20, 10:30** 的缺省值是当前年份当前月份的第 20 日的 10:30:00。日期/时间字段可以用空格或逗号分隔。如果使用空格，则字符串必须用双引号括起。

对于没有世纪字段的 2 位数年份输入 (*yy*)，采用以下解释： [**70-99**, **00-69** (1970-1999, 2000-2069)] 。

-r[*rev*] 检索编号小于或等于 *rev* 的最新修订。如果 *rev* 表示一个分支而不是一个修订，则将检索该分支上的最新修订。 *rev* 由一个或多个数字或符号字段组成，这些字段用 **.** 分隔。符号字段的等效数字可通过 **ci -n** 和 **rcs -n** 命令指定（请参阅 *ci*(1) 和 *rcs*(1) ）。

-s*state* 在所选分支上检索状态设置为 *state* 的最新修订。

-w[*login*] 在所选分支上检索由登录名为 *login* 的用户签入的最新修订。如果省略了参数 *login* ，则假定为调用者的登录。

-j *joinlist* 生成一个新的修订，这是向 *joinlist* 添加修订的结果。 *joinlist* 是一个用逗号分隔的数据对（形式为 rev2**:**rev3 ）列表，其中 *rev2* 和 *rev3* 是符号形式或数字形式的修订编号。对于初始的数据对， *rev1* 表示由选项 **-l**、 ...、 **-w** 选择的修订。对于所有其他数据对， *rev1* 表示由前一个数据对生成的修订。（这样，添加一项之后的输出变为下一项的输入）。

对于每个数据对， **co** 添加有关 *rev2* 的修订 *rev1* 和 *rev3* 。这意味着，由 *rev2* 转化为 *rev1* 的所有更改将应用于 *rev3* 的副本。这在 *rev1* 和 *rev3* 是两个分支的末端，并且这两个分支让 *rev2* 作为其共同祖先时尤其有用。如果在同一分支上 *rev1* < *rev2* < *rev3* ，则添加修订会生成一个类似于 *rev3* 的新修订，但是不应用从 *rev1* 到 *rev2* 的所有更改。如果从 *rev2* 到 *rev1* 的更改与从 *rev2* 到 *rev3* 的更改重叠，则 **co** 将输出一条警告，并包含重叠的部分，按以下方式分隔：

<<<<<<<
rev1
=======
rev3
>>>>>>>

对于初始的数据对，可以省略 *rev2* 。缺省值是共同祖先。如果任一参数指明分支，则假定为这些分支上的最新修订。如果提供了 **-l** 选项，则初始的 *rev1* 将被锁定。

关键字替换

嵌入在文本中的、形式为 **$***keyword***$** 和 **$***keyword***:...$** 的字符串，将被替换为形式为 **$***keyword***:** *value $* 的字符串，在后一种形式中 *keyword* 和 *value* 都是下面列出的数据对。可以将关键字嵌入到文字字符串或注释中，以标识修

订。

最初，用户输入 **$***keyword***$** 形式的字符串。在签出时， **co** 将这些字符串替换为 **$***keyword***:** *value $* 形式的字符串。如果将包含后一种形式字符串的修订重新签入，则在下一次签出时这些值字段将被替换。因此，关键字值在签出时将自动更新。

关键字及其相应的值:

$Author$ 签入修订的用户的登录名。

$Date$ 签入修订的日期和时间。

$Header$ 包含 RCS 文件名、修订编号、日期、作者和状态的标准头文件。

$Locker$ 锁定修订的用户的登录名（如果没有锁定，则为空）。

Log 在签入过程中提供的日志消息，其前面是包含 RCS 文件名、修订编号、作者和日期的头文件。现有的日志消息 不被替换，而新的日志消息在 **$Log:...$** 之后插入。这在源文件中累积完整的更改日志时很有用。

$Revision$ 指定给修订的修订编号。

$Source$ RCS 文件的完整路径名。

$State$ 通过 **rcs -s** 或 **ci -s** 指定给修订的状态。

访问控制列表 (ACL)

不应将可选的 ACL 条目添加到 RCS 文件中，因为它们可能会被删除。

诊断信息

检索到的 RCS 文件名、工作文件名和修订编号将写入诊断信息输出中。退出状态总是与最后一个签出的文件有关。如果操作成功，其值为 0；如果操作不成功，其值为 1。

举例

假定当前目录包含一个名为 **RCS** 的子目录，以及一个名为 **io.c,v** 的 RCS 文件。下列每一个命令都可以从 **RCS/io.c,v** 检索最新修订，并将其存储到 **io.c**：

```
co io.c
co RCS/io.c,v
co io.c,v
co io.c RCS/io.c,v
co io.c io.c,v
co RCS/io.c,v io.c
co io.c,v io.c
```

签出 RCS 文件 **foo.c,v** 的 1.1 版:

```
co -r1.1 foo.c,v
```

将 RCS 文件 **foo.c,v** 的 1.1 版签出到标准输出:

co -p1.1 foo.c,v

签出 **foo.c,v** 文件在 1992 年 9 月 18 日的版本:

co -d"09/18/92" foo.c,v

警告

co 命令通过从 RCS 文件名的末尾删除 **,v** 生成工作文件名。如果给定的 RCS 文件名对于 RCS 文件应驻留的文件系统而言太长，则 **co** 将终止并生成错误消息。

无法禁止关键字扩展，除非以不同方式编写。在 **nroff** 和 **troff** 中，是通过将空字符 **\&** 嵌入到关键字中来实现此目的的。

-d 选项在某些情况下会令人感到迷惑，它不接受早于 1970 年的日期。

-j 选项不适用于含有由一个 **.** 组成的行的文件。

RCS 设计为仅与 *text* 文件一起使用。如果试图将 RCS 与非文本（二进制）文件一起使用，则会导致数据损坏。

作者

co 由 Walter F. Tichy 开发。

另请参阅

ci(1)、 ident(1)、 rcs(1)、 rcsdiff(1)、 rcsmerge(1)、 rlog(1)、 rcsfile(4)、 acl(5)、 rcsintro(5)。

名称

col - 过滤反向换行符和退格符

概要

col [**-blfxp**]

说明

col 从标准输入读取并写入到标准输出上。它执行反向换行符（ASCII 代码 **ESC-7**）以及正向和反向半换行符 （ **ESC-9** 和 **ESC-8** ） 隐含的行覆盖行为。对于过滤使用 **nroff .rt** 命令产生的多列输出以及使用 **tbl** 预处理器产生的输出， **col** 特别有用（请参阅 *nroff*(1) 和 *tbl*(1) ） 。

如果指定了 **-b** 选项，则 **col** 假定正使用的输出设备不能退格。在这种情况下，如果两个或更多字符将出现在同一位置，则仅输出读取的最后一个字符。

如果指定了 **-l** 选项，则 **col** 假定输出设备是行式打印机（而不是字符打印机）并删除退格符以构成复合叠印的完整行。它执行生成所需数目的叠印所必需的最少数目的打印操作。（除了行上的最后一个打印操作外都是由回车符 (\r) 分隔的；最后一个打印操作以换行符 (\n) 结束)

虽然 **col** 在其输入中接受半行移动，但是它通常不在输出时实现它们。相反，将在行间出现的文本被移动到下一个较低的完整行边界。此处理方式可以由 **-f** (fine) 选项禁止；在这种情况下， **col** 的输出可能包含正向半换行符 (ESC-9)，但是仍然决不会包含任一种反向行移动。

除非指定了 **-x** 选项，否则 **col** 会在输出时尽可能将空格转换为制表符以缩短打印时间。

col 假定 ASCII 控制字符 SO (\016) 和 SI (\017) 用于表示其他字符集中文本的开始和结束。将记住每个输入字符所属的字符集，而且在输出时根据需要生成 SI 和 SO 字符以确保用正确的字符集打印每个字符。

在输出时，接受的控制字符只有空格、退格符、制表符、回车符、换行符、SI、SO、VT (\013) 和后跟 **7**、 **8** 或 **9** 的 ESC。 VT 字符是完整反向换行符的替代形式，包括它是为了与此类型的某些早期程序兼容。将忽略所有其他非打印字符。

通常， **col** 忽略在其输入中找到的任何无法识别的转义序列；可以使用 **-p** 选项使 **col** 将这些序列作为常规字符输出，这会受到反向行移动导致的套印的影响。使用此选项的效果令人非常失望，除非用户完全知道转义序列的文本位置。

注释

col 接受的输入格式与 **nroff** 使用 **-T37** 或 **-Tlp** 选项产生的输出相匹配。如果最终处理 **col** 输出的是可以解释半行移动的设备，请使用 **-T37** （以及 **col** 的 **-f** 选项），否则使用 **-Tlp** 。

外部语言环境影响

环境变量

LANG 为没有设置或设置为 null（空）的国际化变量提供了缺省值。如果 **LANG** 未设置或设置为 null（空），则会使用缺省值“C”（请参阅 *lang*(5)) ）。如果任一国际化变量中包含无效设置，则 **col** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

LC_ALL 如果设为空字符串值，则会覆盖所有其他国际化变量的值。

LC_CTYPE 用于确定将文本解释为单字节字符和（或）多字节字符，将字符分类为可打印字符，以及与正则表达式中字符类表达式相匹配的字符。

LC_MESSAGES 用于确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息以及写入到标准输出的信息消息的格式和内容。

NLSPATH 用于确定消息目录的位置，以便处理 **LC_MESSAGES**.

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

col 最常与 **nroff** 和 **tbl** 一起使用。常见用法如下所示：

tbl *filename* **| nroff -man | col | more -s**

（与通常的 *man*(1) 命令非常类似）。此命令允许为表打印竖线和外框。通过 **tbl** 预处理器运行文件，然后由 **nroff** 处理输出，使用 **-man** 宏设置输出格式。接着由 **col** 处理已设置格式的输出，这将设置竖线并对齐文件中的列。最后，文件由 **more** 命令处理，将输出显示在屏幕上，用下划线和突出显示替换斜体和粗体。 **-s** 选项从输出中删除多余的空格以免将多个空行显示在屏幕上。

警告

备份的行数不能超过 128。不能跨页边界备份。

对于每行所包含字符（包括退格符和叠印）的最大数目有限制。最大字符数限制至少为 800 个字符。

将忽略会导致在文档的首行上方备份的本地竖直移动。因此，首行不得包含任何上标。

X/Open 标准中可能会撤消此命令。使用此命令的应用程序可能无法移植到其他供应商的系统上。

另请参阅

nroff(1)、 tbl(1)、 ul(1)、 man(5)。

符合的标准

col：SVID2、SVID3、XPG2、XPG3

名称

comb - 组合 SCCS delta 版

概要

comb [**-p** *SID*] [**-c***list*] [**-o**] [**-s**] *file* ...

说明

comb 命令会生成一个 Shell 程序（请参阅 *sh*(1) ），在运行该程序时将重新构建给定的 SCCS 文件。重新构建的文件通常小于原始文件。可以以任何顺序指定参数，但所有的选项都可应用于所有已命名的 SCCS 文件。如果已命名目录，则执行 **comb** 时，就好像目录中的每个文件都作为已命名文件而指定，但是其中非 SCCS 文件（路径名的最后组成部分不以 **s.** 开头的文件）和不可读的文件将被忽略，而不没有任何提示。如果给定名称 **-** ，则读取标准输入；并将标准输入的每一行解释为要处理的 SCCS 文件的名称；非 SCCS 文件和不可读文件将被忽略，而没有任何提示。生成的 Shell 程序将写到标准输出中。

选项

comb 可识别下列选项。可将每个选项解释为好像只有一个已命名的文件要被处理，但任何选项的作用单独应用于每个已命名的文件。

-p*SID* 要保留的最早的 delta 版的 *SCCS ID* 标识字符串 (SID) 。所有较早的 delta 版都会在已重新构建的文件中被忽略。

-c*list* 要保留的 delta 版的 *list* （有关 *list* 的语法，请参阅 *get*(1) ）。忽略所有其他的 delta 版。

-o 对于每个生成的 **get -e** ，有了该选项，就可以在要创建的 delta 版的发行版中访问已重新构建的文件，否则，会访问最近祖先的重新构建的文件。使用 **-o** 选项可以减小重新构建的 SCCS 文件的大小。它还可以改变原始文件的 delta 版的树的形状。

-s 该选项可使 **comb** 生成一个 Shell 程序，运行该程序时，会生成每个文件的报告：文件名、组合后的大小（以块计）、原始文件的大小（也以块计，以及由下列公式所计算的更改量百分比：

100 ×（原始大小 − 组合后大小）/原始大小

建议在实际对任何 SCCS 文件进行组合之前使用该选项，以确切确定通过组合过程能够节省多少空间。

如果不指定任何选项，则 **comb** 仅保留叶 delta 版，以及为保留树所需要的祖先的最小数量。

外部语言环境影响

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

使用 *sccshelp*(1) 获取解释信息。

举例

命令:

comb -c1.1,1.3,1.6 s.document > save_file

创建名为 **save_file** 的 Shell 脚本，如果执行该脚本，则仅通过使用旧的 **s.document** 中的 delta 版 **1.1** 、 **1.3** 和 **1.6**，就可以创建一个新的 **s.document** 。该脚本将覆盖旧的 **s.document** ；因此，就需要在其他地方复制原始文件。下面是典型技术的示例:

cp s.document s.save
comb -c1.1,1.3,1.6 s.document > save_file
sh save_file

警告

comb 可能重新排列 delta 版的树的形状。组合文件可能节省空间，也可能不节省空间；事实上，重新构建的文件的大小有可能在实际上大于原始文件。

文件

s.BOMB?????	临时文件
comb?????	临时文件

另请参阅

admin(1)、 delta(1)、 get(1)、 sccshelp(1)、 prs(1)、 sh(1)、 sccsfile(4)。

名称

comm - 选择或拒绝两个排序文件的公共行

概要

comm [**-**[**123**]] *file1 file2*

说明

comm 读取 *file1* 和 *file2* ，这两个文件应按增序顺序排序（请参阅 *sort*(1) 和下面的“环境变量”部分），并生成三列输出结果：

第 1 列：　仅出现在 *file1* 中的行，
第 2 列：　仅出现在 *file2* 中的行，
第 3 列：　在两个文件中都出现的行。

如果将 *file1* 或 *file2* 指定为 **-** ，则将使用标准输入。

选项 1、2 或 3 可禁止输出相应的列。因此 **comm -12** 只输出两个文件中都存在的行； **comm -23** 只输出在第一个文件而不在第二个文件中出现的行； **comm -123** 不输出任何行。

外部语言环境影响

环境变量

LC_COLLATE 用于确定 **comm** 希望输入文件采用的排序顺序。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_MESSAGES** 或将其设置为空字符串，则 **LANG** 的值将确定显示的消息所用的语言。如果未在环境中指定 **LC_COLLATE** 或将其设置为空字符串，那么 **LANG** 的值将用作缺省值。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而不是 **LANG** 。如果任一国际化变量包含无效设置，则 **comm** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

下列示例假设 **file1** 和 **file2** 已经按 **LC_COLLATE** 或 **LANG** 环境变量定义的排序顺序进行了排序。

输出 **file1** 和 **file2** 的所有公共行（即，输出列 3）：

comm -12 file1 file2

输出出现在 **file1** 中但不在 **file2** 中的所有行（即，输出列 1）：

comm -23 file1 file2

输出出现在 **file2** 中但不在 **file1** 中的所有行（即，输出列 2）：

comm -13 file1 file2

另请参阅

cmp(1)、 diff(1)、 sdiff(1)、 sort(1)、 uniq(1)。

符合的标准

comm：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

command - 执行简单命令

概要

command *command_name* [*argument*]...

说明

command 使 Shell 将参数视为简单命令，从而禁止查找 Shell 函数。

如果 *command_name* 不是函数名，则 **command** 的功能与省略了 *command* 的功能相同。

操作数

command 可识别下列操作数：

command_name HP-UX 命令名或内置 Shell 命令名。

argument 将解释为 *command_name* 的参数的一个或多个字符串。

需要使用 **command** 命令允许与某一命令同名的函数调用该命令（而不是递归调用函数）。

command 中的描述不暗示对该命令行的分析与其他简单命令有任何不同。例如，

command *a* **|** *b* **;** *c*

以常规方式解释，即将 **|** 或 **;** 视为管道操作符或分号，也不会阻止对 *b* or *c* 的函数查找。

外部语言环境影响

环境变量

PATH 用于在命令搜索过程中使用的搜索路径。

返回值

command 退出时返回下列值之一：

- 如果 **command** 失败：

  126 找到了 *command_name* 指定的实用程序，但不能执行。

  127 **command** 实用程序出现错误，或未找到 *command_name* 指定的实用程序。

- 如果 **command** 未失败：

  command 的退出状态与参数指定的简单命令的退出状态相同：

  command_name[*argument*]...

举例

创建 **cd** 命令的一个版本，使用该命令时总是会输出新工作目录名：

```
cd() {
     command "$@" >/dev/null
     pwd
```

```
}
```

改变上面重新定义的 **cd** 命令，使其在不输出新工作目录名的情况下更改目录：

command cd

另请参阅

getconf(1)、 sh-posix(1)、 confstr(3C)。

符合的标准

command：XPG4、POSIX.2

名称

compact、uncompact、ccat - 压缩和解压缩文件，并显示这些文件的内容

概要

compact [*name*]...

uncompact [*name*]...

ccat [*file*]...

说明

compact 命令可使用自适应 Huffman 编码来压缩指定的文件。如果不指定文件名，则会压缩标准输入，并将其发送到标准输出中。 **compact** 可作为联机算法运行。每次读取一个字节后，就会根据当前前缀码立即对该字节进行编码。对于目前的频率设置来说，该前缀码是最理想的 Huffman 编码。无需在压缩的文件前附加解码树，因为编码器和解码器具有相同的初始状态并保持同步运行。此外， **compact** 和 **uncompact** 可以用作过滤器。尤其是，

... | **compact** | **uncompact** | ...

作为一个空操作运行（速度非常慢）。

当指定了 *file* 参数时，将对该文件进行压缩，并将压缩后的文件放在 *file*.**B** 中，此时 *file* 会被取消链接。压缩文件的前两个字节表示已对文件进行了压缩。这些字节可用于禁止对文件进行重新压缩。

预期的压缩量取决于正在压缩的文件的类型。各种文件在压缩后，其一般的压缩百分比分别为：文本文件，38%；Pascal 源文件，43%；C 源文件，36%；二进制文件，19%。

uncompact 将经过 **compact** 压缩的文件恢复为原始文件。如果不指定文件名，则解压缩标准输入并将结果发送到标准输出中。

ccat 根据由 **compact** 压缩的文件显示其原始文件的内容，而不会解压缩该文件。

访问控制列表 (ACL)

在实现访问控制列表的系统上，当由调用方的有效用户和组 ID 创建了一个新文件时，在更改了原始文件的 ACL 以反映任何所有权改变之后，该 ACL 会被复制到新文件中（请参阅 *acl*(5) 和 *aclv*(5)）。在 JFS 文件系统中，由 **compact** 、 **uncompact** 或 **ccat** 创建的文件不继承其父目录的缺省 ACL 条目（如果有的话），而是保留其原始的 ACL。当正在被压缩或解压缩的文件位于 JFS 文件系统中，而已压缩或解压缩的文件位于 HFS 文件系统中（或者正好相反）时，使用 **ccat** 或作为过滤器使用 **compact** 或 **uncompact** 的结果是，可选的 ACL 条目丢失。

警告

在短文件名系统中，文件名的最后一段必须包含 12 个或更少的字符，以便为追加的 **.B** 留出空间。

相关内容

NFS

网络文件的访问控制列表条目将被汇总（如通过 **stat()** 在 **st_mode** 中返回的值），但不会复制到新文件中（请参阅 *stat*(2)）。

作者

compact 由 Colin L. Mc Master 开发。

文件

***.B** 由 compact 命令创建并由 uncompact 命令删除的压缩文件

另请参阅

compress(1)、 pack(1)、 acl(5)、 aclv(5)。

Gallager, Robert G.，“Variations on a Theme of Huffman”，《I.E.E.E. Transactions on Information Theory》，第 IT-24 卷，第 6 期，1978 年 11 月，页码范围： 668 - 674。

名称

compress、uncompress、zcat、compressdir、uncompressdir - 压缩和扩展数据

概要

压缩文件

compress [**-d**] [**-f**|**-z**] [**-z**] [**-v**] [**-c**] [**-V**] [**-b** *maxbits*] [*file* ...]

uncompress [**-f**] [**-v**] [**-c**] [**-V**] [*file* ...]

zcat [**-V**] [*file* ...]

压缩整个目录子树

compressdir [*options*] [*directory* ...]

uncompressdir [*options*] [*directory* ...]

说明

以下命令按指示的那样压缩和解压缩文件和目录子树:

compress 使用自适应 Lempel-Ziv 编码来减小指定 *file* 的大小。如果有可能减小其大小，则每个 *file* 将被替换为同名的新文件，并添加后缀 **.Z** 以指示它是压缩文件。原始的所有权、模式、访问时间和修改时间保持不变。如果未指定 *file* ，或者指定了 **-** ，则将标准输入压缩到标准输出。

uncompress 将压缩 *file* 恢复为原始格式。得到的文件具有原始文件名、所有权和权限，并删除 **.Z** 文件名后缀。如果未指定 *file* ，或者指定了 **-** ，则将标准输入解压缩为标准输出。

zcat 将压缩 *file* 恢复为原始格式，并将结果发送到标准输出。如果未指定 *file* ，或者指定了 **-** ，则将标准输入解压缩为标准输出。

compressdir 前端处理器。以递归方式向下遍历每个指定的 *directory* 子树，并使用 **compress** 压缩 *directory* 中的每个文件。现有文件将被替换为同名的压缩文件，并加上后缀 **.Z**，条件是得到的文件比原始文件小。如果未指定目录，则将压缩应用于从当前目录开始的所有文件。

options 可能包括任何有效的 **compress** 命令选项（它们最终被传递到 **compress**）。要强制压缩所有文件（甚至在结果比原始文件大时），请使用 **-f** 选项。

uncompressdir 与 **compressdir** 相反。将压缩文件恢复为其原始格式。 *options* 可能包括任何有效的 **uncompress** 命令选项（它们最终被传递到 **uncompress** ）。

获得的压缩量取决于输入的大小、每代码的最大位数 (*maxbits*) 和公共子字符串的分布。通常，将诸如源代码或英文之类的文本减小 50-60%。压缩通常比按 Huffman 编码（如 **pack** 中使用的那样）或自适应 Huffman 编码 (**compact**) 实现的更好，且计算所用时间更少。

选项

这些命令识别以下选项（见上面的 “概要”所示的组合）：

-d 解压缩 *file* 。 **compress -d** 等效于 **uncompress** 。

-f 强制压缩 *file* 。这对压缩整个目录是很有用的，即使其中一些文件实际上没有压缩。如果未指定 **-f** 且 **compress** 在前台运行，则将提示用户是否应该覆盖现有文件。

-z 这与 **-f** 选项相同，不同之处是当存在空压缩时，它不强制压缩。

-v 输出消息，说明每个被压缩文件的减小百分比。

-c 强制 **compress** 和 **uncompress** 写入标准输出；不更改任何文件。 **zcat** 的非破坏性行为与 **uncompress -c** 的相同。

-V 输出当前版本并将选项编译到标准错误。

-b *maxbits* 指定 **compress** 算法将使用的最大位数。缺省值为 16，其范围可以是介于 9 和 16 之间的任何整数。

compress 使用经过修改的 Lempel-Ziv 算法。在《 *A Technique for High Performance Data Compression* 》（作者：Terry A. Welch）和《 *IEEE Computer* 》第 17 卷第 6 期（1984 年 6 月）第 8-19 页中对该算法进行了通俗的讲解。文件中的公共子字符串被首先替换为 9 位代码 257 及更大的值。当达到代码 512 时，该算法将切换到 10 位代码，并继续使用更多的位，直至达到 **-b** 标志指定的限制（缺省值为 16）。

在达到 *maxbits* 限制后， **compress** 将定期检查压缩率。如果压缩率增大，则 **compress** 继续使用现有的代码词典。但是，如果压缩率减小，则 **compress** 忽略子字符串表，并从头开始重建它。这样，该算法就可以适应文件的下一个“块”。

请注意，省略了 **uncompress** 的 **-b** 标志，因为在压缩期间指定的 *maxbits* 参数以及幻数是在输出内编码的，以确保既不尝试解压缩随机数据，也不尝试重新压缩已压缩的数据。

访问控制列表

在压缩和扩展数据时， **compress** 保留文件的访问控制列表。

外部语言环境影响

环境变量

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_MESSAGES** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 。

如果任一国际化变量包含无效设置，则 **compress** 、 **uncompress** 和 **zcat** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

这些命令在完成后返回以下值:

0 成功完成。

2 在（尝试）压缩后，最后一个文件更大。

1 出现错误。

诊断信息

Usage: compress [-f|-z] [-dvcV] [-b maxbits] [file ...]

在命令行上指定的选项无效。

Missing maxbits

在 **-b** 的后面必须有 *maxbits* 。

*file***: not in compressed format**

尚未压缩为 **uncompress** 指定的文件。

*file***: compressed with** *xx***bits, can only handle** *yy***bits**

已压缩 *file* 的程序可以处理比此计算机上的压缩代码更大的 *maxbits* 值。用更小的 *maxbits* 值重新压缩该文件。

*file***: already has .Z suffix -- no change**

假定已压缩文件。重命名文件并重试。

*file***: filename too long to tack on .Z**

输出文件名是带有 **.Z** 扩展名的源文件名，对于源文件所驻留的文件系统来说该名称太长。缩短源文件名，然后重试。

file already exists; do you wish to overwrite (y or n)?

如果您希望输出替换现有文件，则键入 **y** 进行响应；否则，键入 **n** 进行响应。

uncompress: corrupt input

检测到 **SIGSEGV** 违反，这通常意味着输入文件已损坏。

Compression: *xx***.***xx***%**

由压缩保存的输入百分比。（仅与 **-v** 相关）

-- not a regular file: unchanged

当输入文件不是常规文件（例如目录）时，它将保持不变。

-- has *xx***other links: unchanged**

输入文件具有不是符号链接的链接，且保持不变。有关详细信息，请参阅 *ln*(1) 。

-- has symbolic links: unchanged

输入文件具有符号链接，且保持不变。有关详细信息，请参阅 *ln*(1) 。

-- file unchanged

压缩操作未实现任何保存。输入保持不变。

举例

压缩名为 **zenith** 的文件，并将压缩信息输出到终端：

compress -v zenith

终端显示屏显示类似如下的行

zenith: Compression: 23.55% -- replaced with zenith.Z

指示压缩文件比原始文件小 23.55%，或者类似如下的行

zenith: Compression: -12.04% -- file unchanged

指示必须将另外 12.04% 的空间用来压缩文件。

通过键入以下任一命令来撤消压缩：

uncompress zenith.Z
compress -d zenith.Z

这会将文件 **zenith.Z** 恢复为其原始的未压缩格式和名称。

如果未指定文件，则 **uncompress** 将对标准输入执行操作。例如，列出已压缩的 tar 文件：

uncompress -c arch.tar.Z | tar -tvf -

警告

虽然压缩文件在具有大内存的计算机之间是兼容的，但是应该将 **-b**12 用于到具有小进程数据空间（64KB 或更小）的体系结构的文件传输。

NFS

网络文件的访问控制列表将被汇总（如通过 **stat()** 在 **st_mode** 中返回的值），但不会复制到新文件中（请参阅 *stat*(2) ）。

作者

compress 由 Joseph M. Orost、Kenneth E. Turkowski、 Spencer W. Thomas 和 James A. Woods 开发。

文件

***.Z** 由 **compress** 创建并由 **uncompress** 删除的压缩文件。

另请参阅

compact(1)、pack(1)、acl(5)。

符合的标准

compress： XPG4

名称

convert - 转换音频文件

概要

/opt/audio/bin/convert [*source_file*] [*target_file*] [**-sfmt** *format*] [**-dfmt** *format*]
[**-ddata** *data_type*] [**-srate** *rate*] [**-drate** *rate*]
[**-schannels** *number*] [**-dchannels** *number*]

说明

该命令可将音频文件从一种支持的文件格式、数据格式、采样速率和通道数转换为另一种。未转换的文件保留为源文件格式。

-sfmt *format* **-dfmt** *format*

源文件和目标文件的文件格式。每一 *format* 可以为下列值之一:

au Sun 文件格式

snd NeXT 文件格式

wav Microsoft RIFF Waveform 文件格式

u MuLaw 格式

al ALaw

l16 线性 16 位格式

lo8 偏移（无符号）线性 8 位格式

l8 线性 8 位格式

如果省略了 **-sfmt** ，则 **convert** 将使用源文件中的头或文件扩展名。如果为目标文件提供了文件扩展名，则可以省略 **-dfmt** 。

-ddata *data_type*

目标文件的数据类型。 *data_type* 可以是下列值之一:

u MuLaw

al ALaw

l16 线性 16 位

lo8 偏移（无符号）线性 8 位数据

l8 线性 8 位数据

如果省略了 **-ddata** ，则 **convert** 将使用适当的数据类型，通常为源文件的数据类型。

-srate *rate* **-drate** *rate*

源文件和目标文件的每秒采样数。典型的采样率范围为 8 至 11k（对于语音质量）以及 8 至 44,100（对于 CD 质量）。可以使用 **k** 表示千。例如， **8k** 表示每秒 8,000 的采样率。

如果省略了 **-srate** ，则 **convert** 将使用由源文件头或其文件扩展名定义的速率。对于不带任何扩展名的原始文件，使用 8,000。通过播放音频文件，可以确定 8,000 采样率是否合适。

如果省略了 **-drate** ，则 **convert** 将使用适合于目标文件格式的采样率；如果可能，匹配源文件的采样率。

-schannels *number* **-dchannels** *number*

源文件和目标文件中的通道数。对于单声道，使用 **1** ；对于立体声，使用 **2** 。如果省略了 **-schannels** ，则 **convert** 将使用头中的信息；对于原始数据文件使用单声道。

如果省略了 **-dchannels** ，则 **convert** 将匹配源文件使用的通道（通过头或 **-schannels** 选项）；对于原始数据文件使用单声道。

举例

将原始数据文件转换为带有头的文件。

```
cd /opt/audio/bin
convert  beep.l16  beep.au
```

在源文件没有任何扩展名、以每秒 11,025 的速率进行采样，且包含立体声数据的情况下，将原始数据文件转换为带有头的文件。

```
cd /opt/audio/bin
convert  beep  beep.au -sfmt l16 -srate 11025 -schannels 2
```

为了节省磁盘空间，可将音频文件从 CD 质量声音转换为语音质量声音。

```
cd /opt/audio/bin
convert idea.au  idea2.au -ddata u -drate 8k -dchannels 1
```

作者

convert 由 HP 开发。

Sun 是 Sun MicroSystems, Inc. 的商标。

NeXT 是 NeXT Computers, Inc. 的商标。

Microsoft 是 Microsoft Corporation 的商标。

另请参阅

audio(5)、 asecure(1M)、 aserver(1M)、 attributes(1)、 send_sound(1)。

《Using Audio Developer’s Kit》

名称

cp - 复制文件和目录子树

概要

cp [**-f**|**-i**] [**-p**] [**-S**] [**-e** *extarg*] *file1 new_file*

cp [**-f**|**-i**] [**-p**] [**-S**] [**-e** *extarg*] *file1* [*file2* ...] *dest_directory*

cp [**-f**|**-i**] [**-p**] [**-S**] [**-R**|**-r**] [**-e** *extarg*] *directory1* [*directory2* ...] *dest_directory*

说明

cp 用于将:

- *file1* 复制到新的或现有的 *new_file* ,
- *file1* 复制到现有的 *dest_directory* ,
- *file1* 、 *file2* ... 复制到现有的 *dest_directory* ,
- 目录子树 *directory1* 复制到新的或现有的 *dest_directory* ，或者
- 多个目录子树 *directory1* 、 *directory2* ... 复制到新的或现有的 *dest_directory* 。

如果 *file1* 和 *new_file* 相同， **cp** 将失败（慎用 Shell 元字符）。当目标是目录时，一个或多个文件将复制到该目录中。如果复制两个或更多个文件，目标必须是目录。将单个文件复制到新文件时，如果存在 *new_file* ，其内容将损坏。

如果目标 *dest_directory* 或现有目标文件 *new_file* 的访问权限禁止写入， **cp** 将异常中止，并生成错误消息“无法创建 *file* ”。

要将一个或多个目录子树复制到其他目录，需要使用 **-r** 选项。将一个文件复制到另一个文件或者将多个文件复制一个目录时，如果使用 **-r** 选项，则会将其忽略。

如果 *new_file* 是指向包含其他链接的现有文件的链接， **cp** 将覆盖现有文件并保留所有链接。如果将一个文件复制到一个现有文件， **cp** 不会更改现有文件的访问权限位、所有者或组。

将文件复制到已经不存在的目录或新文件时， **cp** 将以与 *file1* 相同的文件权限位创建一个新文件，如果未指定 **-p** 选项，该文件权限位将由用户的文件创建掩码进行修改，然后通过 S_IRWXU 逐位取内含或。新文件的所有者和组是该用户的所有者和组。 *new_file* 的上次修改时间（如果不存在 *new_file* ，则是上次访问时间）和源 *file1* 的上次访问时间设置为进行复制的时间。

选项

-i （交互式复制）致使 **cp** 在复制将覆盖现有文件的文件之前将提示写入标准错误并等待响应。如果标准输入的响应是确定的，则将在权限允许的情况下复制该文件。如果同时指定了 **-i** （交互）和 **-f** （强制复制），则将忽略 **-i** 选项。

-f 在复制之前强制删除现有目标路径名，不要求用户确认。该选项具有毁坏并替换任何其名称和目录位置与复制操作所创建的新文件的名称和位置存在冲突的现有文件的效果。

-p （保留权限）致使 **cp** 在权限允许的情况下尽可能多地在副本中保留修改时间、访问时间、文件模式、用户 ID 和组 ID。

-r （递归子树复制）致使 **cp** 将每个源目录下的子树复制到 *dest_directory* 。如果存在 *dest_directory* ，它必须是目录，此时 **cp** 将在 *dest_directory* 中创建一个与 *file1* 同名的目录，然后将 *file1* 下的子树复制到 *dest_directory/file1* 。如果已存在 *dest_directory/file1* ，则将出错。如果不存在 *dest_directory* ， **cp** 将创建该目录，并将 *file1* 下的子树复制到 *dest_directory* 。请注意， **cp -r** 不能合并子树。

通常将复制的常规的文件和目录。如果用户拥有对文件的访问权，则将复制字符特殊设备、块特殊设备、网络特殊设备、命名管道、符号链接和套接字；否则，将输出警告声明无法创建该文件，并跳过该文件。

dest_directory 不应该驻留在 *directory1* 中， *directory1* 也不应该具有循环目录结构，否则在这两种情况下 **cp** 都将复制无限数量的数据库。

在 **UNIX95** 标志集下，如果将多个源复制到不存在的目录， **cp** 将退出并报错。

-R （递归子树复制） **-R** 选项等效于 **-r** 选项。

使用 **-R** 和 **-r** 选项，除了常规文件和目录， **cp** 还将复制 FIFO、字符和块设备文件以及符号链接。只有超级用户才能复制设备文件。其他所有用户都将收到错误。符号链接的复制将使得目标指向源所指的相同位置。

警告：复制包含设备特殊文件的目录树时，请使用 **-r** 选项；否则，将从设备特殊文件中读取无限数量的数据，并将这些数据复制为在目标目录中占用大量文件系统空间的特殊文件。

-e *extarg*

指定如何处理要复制的文件的任何范围属性。 *extarg* 采用下列值之一。

warn 如果无法复制范围属性，则发出一条警告消息，但仍会复制该文件。
ignore 不复制范围属性。
force 如果无法复制范围属性，则无法复制文件。

如果将文件复制到不支持范围属性的文件系统，或者该文件系统与初始文件系统具有不同的块大小，则将无法复制范围属性。如果未指定 **-e** ， *extarg* 的缺省值是 **warn**。

-S 指定“安全”模式。这将尽量减少 **cp** 在使用 **O_DSYNC** 标志打开目标文件时对系统性能的影响。当副本总大小（而不是单个文件）占用系统缓冲区缓存中相当大的一部分， 建议使用该选项。使用该选项将增加复制时间。

访问控制列表 (ACL)

如果 *new_file* 是一个新文件或者新文件是在 *dest_directory* 中创建的，它将继承初始 *file1* 、 *file2* 等的访问控制列表，该列表会经过更改，以反映两个文件之间任何的所有权差异（请参阅 *acl*(5) 和 *aclv*(5) ）。在 JFS 文件系统中，由 **cp** 创建的新文件不继承其父目录的缺省 ACL 条目（如果有），而保留所复制文件的 ACL。将文件从 JFS 文件系统复制到 HFS 文件系统或反向复制时，将丢失可选的 ACL 条目。

外部语言环境影响

环境变量

UNIX95 指定该命令使用 XPG4 行为。

LC_CTYPE 确定将文本解释为单字节和（或）多字节字符的方法。

LANG 和 **LC_CTYPE** 确定 y（用于"是或否"查询）的本地语言等效表示法。

LANG 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果 **LANG** 未指定或设置为空字符串，则将使用缺省的"C"而不是 **LANG** （请参阅 *lang*(5)）。如果任一国际化变量中包含无效设置， **cp** 会假定所有国际化变量都缺省为"C"。请参阅 *environ*(5)。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

以下命令将目录 *sourcedir* 及其内容移到文件系统中的新位置 (*targetdir*)。由于 **cp** 创建新的目录，因此目标目录 *targetdir* 不应存在。

cp -r *sourcedir targetdir* **&& rm -rf** *sourcedir*

-r 选项将目录 *sourcedir* 中的子树（文件和子目录）复制到目录 *targetdir*。双 & 号 (**&&**) 会导致条件性操作。如果 **&&** 左侧的操作成功，则将执行右侧的操作（删除旧目录）。如果 **&&** 左侧的操作不成功，则不删除旧目录。

该示例等效于:

mv *sourcedir targetdir*

要当前目录中的所有文件和目录子树复制到现有的 *targetdir*，请使用:

cp -r * *targetdir*

要将 *sourcedir* 中的所有文件和目录复制到 *targetdir*，请使用:

cp -r *sourcedir***/*** *targetdir*

请注意，目录路径名可以位于 *sourcedir* 和 *targetdir* 之前。

要创建长度为零的文件，请使用下列任一命令:

cat /dev/null > *file*
cp /dev/null *file*
touch *file*

相关内容

NFS

网络文件的访问控制列表将被汇总（如通过 **stat()** 在 **st_mode** 中返回的值），但不会复制到新文件中。对这些文件使用 **mv** 或 **ln** 时，如果请求覆盖文件的权限，将不会在模式值后输出 **+**。

作者

cp 由 AT&T、加州大学伯克利分校和 HP 联合开发。

另请参阅

cpio(1)、 ln(1)、 mv(1)、 rm(1)、 link(1M)、 lstat(2)、 readlink(2)、 stat(2)、 symlink(2)、 symlink(4)、 acl(5)、 aclv(5)。

符合的标准

cp：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

cpio - 向内和向外复制文件归档；复制目录树

概要

cpio -o [**-e** *extarg*] [**achvxABC**]

cpio -i[**bcdfmrstuvxBPRSU6**] [*pattern*...]

cpio -p [**-e** *extarg*] [**adlmruvxU**] *directory*

说明

cpio 命令在磁带、其他设备或常规文件上保存和恢复文件归档，并在复制目录树结构的同时，将文件从一个目录复制到另一个目录。 **cpio** 处理完文件后将报告写入的块数。

cpio -o （向外复制，导出） 读取标准输入以获得路径名的列表，并将那些文件与路径名和状态信息一起复制到标准输出。按 512 字节边界填充输出。

cpio -i （向内复制，导入） 从标准输入提取文件，假定标准输入是以前 **cpio -o** 的结果。

如果指定了 *pattern*... ，则仅选择根据模式匹配表示法（请参阅 *regexp*(5) ）的规则，其名称与 *pattern* 匹配的文件。模式上的前导 **!** 指示仅应该选择与模式的剩余部分 不匹配的那些名称。可以指定多个模式。模式是累加性的。如果未指定 *pattern* ，则缺省值为 ***** （选择所有文件）。另请参阅 **f** 选项。

提取的文件是有条件创建的，并被复制到当前目录树，如下面所述的选项确定的那样。在 **cpio -o** 创建归档时，除非使用 **U** 选项，否则文件的权限与原始文件的权限相匹配。文件所有者和组是当前用户的，除非用户具有相应的特权，在这种情况下， **cpio** 保留以前 **cpio -o** 的文件所有者和组。

cpio -p （通过） 读取标准输入以获得文件路径名的列表，然后有条件地创建这些文件并将其复制到目标 *directory* 树，如下面所述的选项确定的那样。 *directory* 必须存在。将目标路径名解释为相对 *directory* 。

如果指定了 **-p** 选项，则在处理链接时仅传递链接，实际上不读取或写入数据块。这对于 **cpio -pl** 尤其值得注意，因为很可能将所有文件都创建为链接，这样就不会写入块且 **cpio** 报告“0 个块”（有关 **-l** 选项的说明，请参阅下文）

选项

cpio 识别以下选项，可以根据需要将它们追加到 **-i** 、 **-o** 和 **-p** 。在这些选项与 **-i** 、 **-o** 或 **-p** 之间不允许有空格和连字符。

a 在复制输入文件后，重置其访问时间。

b 同时交换字节和半字。仅与 **-i** 一起使用。有关详细信息，请参阅 **P** 选项；另请参阅 **s** 和 **S** 选项。

c 以 ASCII 字符格式写入或读取头信息，以实现可移植性。

d 根据需要创建目录。

-e *extarg* 指定如何处理要归档或复制的文件的任何盘区属性。 *extarg* 采用以下值之一。

warn 归档或复制文件，并在无法保留盘区属性时，发出警告消息。

ignore 不发出警告消息，即使无法保留盘区属性。

force 将不归档具有盘区属性的任何文件，但发出警告消息。

在使用 **-o** 选项时，在归档中不保留盘区属性。此外，如果要将文件复制到不支持盘区属性的文件系统，则 **-p** 选项将不保留盘区属性。如果未指定 **-e** ，则 *extarg* 的缺省值是 **warn** 。

f 向内复制除 *pattern*... 选择的那些文件之外的所有文件。

h 跟踪符号链接，好像它们是普通的文件或目录。通常， **cpio** 对链接进行归档。

l 尽可能链接文件而不是复制它们。此操作不删除现有文件。仅与 **-p** 一起使用。

m 保留以前的文件修改时间。此选项不影响要复制的目录。

r 以交互方式重命名文件。如果用户键入空行，则将跳过该文件。

s 交换文件的所有字节。仅与 **-i** 一起使用。有关详细信息，请参阅 **P** 选项；另请参阅 **s** 和 **S** 选项。

t 仅打印输入的目录。不创建、读取或复制文件。

u 无条件复制（通常，旧文件不会替换同名的新文件）。

v 在处理文件名的同时打印它们的列表。当与 **t** 选项一起使用时，目录的格式如下：

numeric-mode owner-name blocks date-time filename

其中 *numeric-mode* 是数字格式的文件特权， *owner-name* 是文件所有者的名称， *blocks* 是文件的大小（以 512 字节的块为单位）， *date-time* 是上次修改文件的日期和时间， *filename* 是在归档中记录的文件路径名。

x 保存或恢复设备专用文件。由于 **mknod()** 用于在恢复时重新创建这些文件，因此 **-ix** 和 **-px** 只能由具有相应特权的用户使用（请参阅 *mknod*(2) ）。此选项仅供系统内（备份）使用。从操作系统的早期版本或者从不同的系统恢复设备文件可能是非常危险的。 **cpio** 可能阻止从归档恢复某些设备文件。

A 禁止有关可选的访问控制列表条目的警告消息。 **cpio** 不备份文件访问控制列表中的可选访问控制列表条目（请参阅 *acl*(5) ）。通常，对于具有可选的访问控制列表条目的每个文件，都将输出警告消息。

B 按 5120 字节输入/输出到记录的块（不适用于 **cpio -p** ）。此选项仅对定向到支持可变长度记录的设备（如磁带）或从其定向的数据有意义。

C 使 **cpio** 在每个卷的起始处对自身执行检查点操作。如果 **cpio** 在启用即时报告模式的情况下写入流式磁带机，且出现写入错误，则它通常会异常中止并以返回代码 **2** 退出。如果指定此选项，则 **cpio** 将改为从检查点自动重新启动自身，并重写当前卷。或者，如果 **cpio** 没有写入这样的设备但出现了写入错误，则 **cpio** 通常会继续执行下一个卷。但是，如果指定此选项，则用户可以选择忽略错误还是重写当前卷。

P 读取在未使用 **c** 选项的情况下，在 PDP-11 或 VAX 系统（带有字节交换）上写入的文件。仅与 **-i** 一起使用。不更改在此模式下复制的文件。非 ASCII 文件很可能需要进行进一步处理才能是可读取的。此处理通常需要知道文件内容，因此无法始终由此程序完成。当适合于交换磁带上（而不仅仅是磁带头中）的所有字节时，可以使用 **b** 、 **s** 和 **S** 选项。通常，最好使用 **P** 处理文本，使用其他选项之一处理二进制数据。

R 在 **cpio** 变成“不同步”时会自动重新同步（请参阅“诊断信息”一节）。

S 交换文件中的所有半字。仅与 **-i** 一起使用。有关详细信息，请参阅 **P** 选项；另请参阅 **b** 和 **s** 选项。

U 使用进程的文件模式创建掩码（请参阅 *umask*(2) ），按与 *creat*(2) 相同的方式修改所创建文件的模式。

6 处理 UNIX 第六版格式文件。仅与 **-i** 一起使用。

请注意，使用原始设备文件创建的 **cpio** 归档必须使用原始设备文件进行读取。

当到达磁带末尾时， **cpio** 将提示用户输入新的专用文件，然后继续。

如果要将一个或多个元字符传递到 **cpio** 但不让 Shell 扩展它们，请确保在每个元字符前面加上反斜杠 (\) 。

使用 **-ox** 选项写入的设备文件（如 **/dev/tty03** ）不传输到 HP-UX 的其他实现。

外部语言环境影响

环境变量

LC_COLLATE 用于确定在评估文件名生成的模式匹配表示法中使用的排序序列。

LC_CTYPE 用于确定将文本解释为单字节字符和（或）多字节字符，以及按模式匹配表示法与字符类表达式匹配的字符。

LC_TIME 用于确定在使用 **v** 选项列出归档的内容时输出的日期和时间字符串的格式和内容。

LANG 用于确定显示消息的语言。

如果未在环境中指定 **LC_COLLATE**、 **LC_CTYPE**、或 **LC_TIME**，或者将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG**。如果任一国际化变量包含无效设置，则 **cpio** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

cpio 返回以下退出代码:

0 成功完成。查看文件中无法传输的标准错误。

1 重新同步期间出现错误。一些文件可能尚未恢复。

2 不同步错误。文件头已损坏或格式错误。

诊断信息

Out of phase--get help
Perhaps the "c" option should[n't] be used

cpio -i 无法读取归档文件的头。头已损坏或者是以其他格式写入的。如果不指定 **R** 选项，则 **cpio** 返回退出代码 **2** 。

如果尚未显示文件名，则问题可能出在格式上。请尝试指定其他的头格式选项： null（空）表示标准格式； **c** 表示 ASCII； **b** 、 **s** 、 **P** 或 **S** 表示字节交换格式之一； **6** 表示 UNIX 第六版。

否则，头可能已损坏。使用 **R** 选项使 **cpio** 尝试自动重新同步文件。重新同步意味着， **cpio** 尝试在归档文件中查找下一个好的文件头并从该处继续处理。如果 **cpio** 尝试从不同步状态重新同步，则它将返回退出代码 **1** 。

其他诊断消息是自述性的。

举例

将目录的内容复制到磁带归档:

ls | cpio -o > /dev/rmt/c0t0d0BEST

复制目录层次结构:

cd olddir
find . -depth -print | cpio -pd *newdir*

通常的情况

find . -depth -print | cpio -oB >/dev/rmt/c0t0d0BEST

可以通过以下项进行更有效的处理:

find . -cpio /dev/rmt/c0t0d0BEST

警告

由于行业标准和互操作性目标， **cpio** 不支持归档大于 2 GB 的文件或具有大于 60 K 的用户（或组）ID 的文件。其用户（或组）ID 大于 60 K 的文件将在当前进程的用户（或组）ID 下进行归档和恢复。

如果指定的 **cpio** 归档文件与属于该 **cpio** 归档的原始文件驻留在同一目录中，则不要将 **cpio** 的输出重定向到该归档文件。这可能会导致数据丢失。

cpio 删除传输给它的文件名列表中的任何前导 **./** 字符。

路径名不能超过 **PATH_MAX** 个字符（请参阅 **<limits.h>** 和 *limits*(5)）。如果存在太多的唯一链接文件，则程序将因内存不足而无法跟踪它们。因此，将丢失链接信息。只有具有相应特权的用户才可以复制专用文件。

使用 **-ox**[**c**] 选项在 HP 计算机上写入的 **cpio** 磁带有时可能会误导不支持 **x** 选项的 **cpio** （非 HP）版本。如果 **cpio** 的非 HP（或非 AT&T）版本碰巧被修改，以便 (HP) **cpio** 将它识别为设备专用文件，则可能会创建伪设备文件。

如果无法访问 **/dev/tty**，则 **cpio** 将发出消息并退出。

-pd 选项不创建在命令行上键入的目录。

-idr 选项不创建空目录。

-plu 选项不将文件链接到现有文件。

POSIX 将名为 **TRAILER!!!** 的文件定义为归档结束标志。因此，如果在由 **cpio -o** 写入的文件组中包含该名称的文件，则将该文件解释为归档结束标志，且不复制其余文件。建议做法是避免将文件命名为类似归档结束标志文件名的任何名称。

要创建符合 POSIX 的 **cpio** 归档，必须使用 **c** 选项。要读取符合 POSIX 的 **cpio** 归档，必须使用 **c** 选项，而且不应该使用 **b**、 **s**、 **S** 和 **6** 选项。如果用户没有相应的特权，则在读取归档时还必须使用 **U** 选项以获取符合 POSIX 的行为。具有相应特权的用户不应该使用此选项来获取符合 POSIX 的行为。

相关内容

如果为 **cpio** 提供的路径将符号链接作为最后一个元素包含在内，则将遍历此链接，且路径名解析继续进行。 **cpio** 使用符号链接的目标，而不是该链接的目标。

另请参阅

ar(1)、 find(1)、 tar(1)、 cpio(4)、 acl(5)、 environ(5)、 lang(5)、 regexp(5)。

符合的标准

cpio： SVID2、SVID3、XPG2、XPG3

名称

cpp - C 语言预处理器

概要

/usr/ccs/lbin/cpp [*option* ...] [*ifile* [*ofile*]]

说明

cpp 是 C 语言预处理器，它使用 **cc** 命令进行第一遍 C 语言编译（请参阅 *cc*(1)）。其目的是处理 **#include** 以及条件编译指令和宏。因此，**cpp** 的输出设计为采用可作为 C 编译器下一次通过的输入来接受的形式。随着 C 语言的演变，**cpp** 和 C 编译包的剩余部分将进行修改，以遵循这些变化。因此，不建议在本框架之外的框架中使用 **cpp**。调用 **cpp** 的首选方式是通过 **cc** 命令，这是因为 **cpp** 的功能可能会在将来移到他处。有关一般性宏处理器的信息，请参阅 *m4*(1)。

cpp 选择性地接受两个文件名作为参数。*ifile* 和 *ofile* 分别是预处理器的输入和输出。如果未指定，它们将缺省为标准输入和标准输出。

选项

cpp 可识别下列选项：

-A 删除所有以一个字母和 **_HPUX_SOURCE** 开头的预定义符号。用户在使用该选项时应该定义 **_POSIX_SOURCE** 或 **_XOPEN_SOURCE**。

-C 缺省情况下，**cpp** 将删除 C 样式注释。如果指定了 **-C** 选项，则将传递所有注释（**cpp** 指令行上的注释除外）。

-D_name_

-D_name_**=**_def_ 定义 *name*，就像是通过 **#define** 指令一样。如果未给定 **=**_def_，*name* 将定义为 **1**。**-D** 选项的优先权低于 **-U** 选项。因此，如果在 **-U** 选项和 **-D** 选项中使用相同的名称，则无论这两个选项采用怎样的顺序，该名称将是未定义的。

-H_nnn_ 将内部宏定义表的大小更改为 *nnn* 字节。缺省的缓冲区大小至少是 8 188 字节。该选项用于消除“宏参数太大”、“宏调用太大”、“宏参数在替换后太大”、“引用的宏参数太大”、“宏缓冲区太小”、“输入行太长”和“连接的输入行太长”错误。

-h[*inclfile*] 生成包括文件并将结果发送到文件 *inclfile*。如果省略了参数 *inclfile*，结果将发送到标准错误。

-I_dir_ 更改搜索其名称不以 **/** 开头的 **#include** 文件的算法，以便在查看标准列表中的目录之前查看 *dir*。因此，将首先在包含 **#include** 行的文件的目录中，然后按从左到右的顺序在 **-I** 选项命名的目录中，最后在标准列表上的目录中，搜索其名称在双引号内的 **#include** 文件 (**" "**)。对于其名称在尖括号 (**<>**) 内的 **#include** 文件，将不搜索包含 **#include** 行的文件的目录。但仍会搜索目录 *dir*。

-M[*makefile*] 生成生成文件的相关性，并将结果发送到文件 *makefile*。如果省略了参数 *makefile*，结果将发送到标准错误。

-P 对输入进行预处理，而不生成行控制信息供 C 编译程序下次通过使用。

-T HP-UX 不再将预处理器符号限制为八个字符。 **-T** 选项强制 **cpp** 仅使用前八个字符来区别不同的预处理器名称。就名称程度而言，该行为与其他某些系统上的预处理器相同，它为进行向后兼容而提供的。

-U*name* 删除 *name* 的任何初始定义，其中 *name* 是由特定预处理器预定义的保留符号。这些符号的当前列表包括：

操作系统：	**unix** **__unix**
硬件：	**hp9000s200** **hp9000s300** **__hp9000s300** **hp9000s500** **hp9000s800** **__hp9000s800** **hp9000ipc** **hppa** **__hppa** **_PA_RISC1_0** **_PA_RISC1_1** **_SIO** **_WSIO**
UNIX 系统变型：	**hpux** **__hpux** **_HPUX_SOURCE** **PWB** **_PWB**
lint(1)	: **lint** **__lint**

此外将保留所有以一个下划线和一个大写字母（或另一个下划线）开头的字符。其他符号可能由 C 编译程序的 **CCOPTS** 变量或其他命令行选项在编译时定义（请参阅 *cc*(1)）。所有 HP-UX 系统均定义了符号 **PWB** 、 **hpux** 、 **unix** 、 **_PWB** 、 **__hpux** 和 **__unix** 。每个系统会至少定义一种硬件变型（如果可行）。 lint 符号在 *lint*(1) 运行时定义。请参阅相关内容。

cpp 可以理解两个特殊的名称。 **__LINE__** 定义为由 **cpp** 计数的当前行号（十进制数字）。 **__FILE__** 定义为 **cpp** 所知的当前文件名（C 字符串）。与其他任何定义名称相同，它们可以在任意位置（包括在宏中）使用。

指令

所有 **cpp** 指令的开头均是以 **#** 开头的行。 **#** 和指令之间可以使用任意数量的空白字符和制表符。指令包括：

#define *name token-string*

将 *name* 的后继实例替换为 *token-string* 。 *token-string* 可以为空。

#define *name*(*arg*, ... , *arg*) *token-string*

将后接 **(** 的 *name* 的后继实例、逗点分隔的参数集列表和 **)** 替换为 *token-string* ，其中 *arg* 在 *token-string* 中的每个实例均替换为逗点分隔列表中的相应标记集。当带有参数的宏进行扩展时，参数将按原样放入扩展的 *token-string* 。扩展整个 *token-string* 后， **cpp** 将重新开始扫描在新创建的 *token-string* 开头扩展的名称。

请注意， *name* 和 **(** 之间不能使用空格。

#endif [*text***]** 结束以测试指令（ #if 、 **#ifdef** 或 **#ifndef** ）开头的行部分。每个测试指令必须有匹配的 **#endif**。与 **#endif** 在同一行上出现的任何 文本将被忽略，因此可用于标记匹配的 **#if**-**#endif** 对。这样在读取源时将更容易将 **#if** 、 **#ifdef** 和 **#ifndef** 指令与其关联的 **#endif** 指令相匹配。

#elif *constant-expression*

等价于：

#else
#if *constant-expression*

#else 取匹配该指令的测试指令的相反之意。因此，如果忽略该指令之前的行，之后的行将出现在输出中，反之亦然。

#if *constant-expression*

当且仅当 *constant-expression* 计算结果为非零时，后面的行才出现在输出中。所有二元非赋值 C 运算符、 **?:** 运算符、一元 **-** 、 **!** 和 **~** 运算符在 *constant-expression* 中均是合法的。运算符的优先顺序与 C 语言所定义的相同。

还有一个一元运算符 **defined** ，它可以通过以下两种形式在 *constant-expression* 中使用：**defined(***name***)** 或 **defined** *name* 。这样就可以在 **#if** 指令中使用 **#ifdef** 和 **#ifndef** 。

只有 **cpp** 所知的运算符、整数常量和名称才应在 *constant-expression* 中使用。尤其是， **sizeof** 运算符不可用。

#ifdef *name* 当且仅当 *name* 已成为先前 **#define** 的主题，而不是其间 **#undef** 的主题时，后面的行才出现在输出中。

#ifndef *name* 当且仅当 *name* 已成为先前 **#define** 的主题，而不是其间 **#undef** 的主题时，后面的行才不出现在输出中。

#include "*filename***"**
#include <*filename***>**

在此处包括 *filename* 的内容（然后通过 **cpp** 运行）。有关详细信息，请参阅上面的 **-I** 选项。

#line *integer-constant* **"***filename***"**

可使 **cpp** 为 C 编译程序下次通过生成行控制信息。 *integer-constant* 是下一行的行号， *filename* 是它所来自的文件。如果省略了 *filename* 和引号，当前文件名将保持不变。

#undef *name* 可使从现在开始忘记 *name* 的定义（如果有）。

测试指令和可能的 **#else** 指令可以进行嵌套。 **cpp** 支持长度最大 255 个字符的名称。

注释

宏替换方案已更改。先前版本的 **cpp** 将宏保存在宏定义表中，该表的缺省大小是 128 000 字节。当前版本的 **cpp** 将该宏定义表替换为几个小缓冲区。小缓冲区的缺省大小是 8 188 字节。

外部语言环境影响

环境变量

LC_CTYPE 确定如何将注释和字符串文字解释为单字节和（或）多字节字符。

LANG 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5)）。如果任一国际化变量包含无效设置，则 **cpp** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5)。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

cpp 生成的错误消息应该是自述性消息。出错处的行号和文件名将随诊断信息一起输出。

警告

在参数列表中找到用于扩展宏的换行符时，先前版本的 **cpp** 会在找到并扩展这些换行符时将其关闭。当前版本的 **cpp** 会将这些换行符替换为空白字符，以减轻在出现这种情况时先前版本存在的问题。

相关内容

工作站

符号 **hp9000s700** 和 **__hp9000s700** 不是 **-U** 选项可识别的保留字符。它们由编译程序自动（或通过使用编译程序选项）提供给 **cpp** 。例如，在 Series 700 系统，以下命令:

cc -v t.c

生成:

/usr/ccs/lbin/cpp t.c /var/tmp/ctmAAAa29220 -D__hp9000s700 -D__hp9000s800 ...

（另请参阅 **cc** 命令的 **-D** 选项）。

文件

/usr/include **#include** 文件的标准目录

另请参阅

m4(1)。

符合的标准

cpp：SVID2、SVID3、XPG2

名称

crontab - 用户作业文件调度程序

概要

crontab [*file*]

crontab -e [*username*]

crontab -l [*username*]

crontab -r [*username*]

说明

crontab 命令为用户管理 crontab 文件。您可以使用 crontab 文件调度由 **cron** （请参阅 *cron*(1M) ）定期自动执行的作业。该命令具有以下四种形式：

crontab [*file*] 创建或替换您的 crontab 文件，方法是将指定的 *file* 或标准输入（如果省略了 *file* 或者 - 被指定为 *file* ）复制到 crontab 目录 **/var/spool/cron/crontabs** 中。您的 crontab 文件在 crontab 目录中的名称与您的有效用户名相同。

crontab -e [*username*] 编辑用户的 crontab 文件的副本，或者创建要编辑的空文件（如果 crontab 文件不存在）。在完成编辑后，会将文件复制到 crontab 目录中，作为用户的 crontab 文件。

crontab -l [*username*] 列出用户的 crontab 文件。

crontab -r [*username*]" 从 crontab 目录中删除用户的 crontab 文件。

只有特权用户才能在 **-e** 、 **-l** 或 **-r** 选项的后面使用 *username* ，以编辑、列出或删除指定用户的 crontab 文件。

crontab 文件中的条目是均包含六个字段的行。这些字段由空格或制表符分隔。这些行具有以下格式：

minute hour monthday month weekday command

前五个字段是整数模式，指定应该执行第六个字段 *command* 的时间。它们可以具有以下值范围：

minute 一小时中的分钟 (**0–59**)

hour 一天中的小时 (**0–23**)

monthday 一月中的某天 (**1–31**)

month 一年中的某月 (**1–12**)

weekday 一周中的某天 （ **0–6**，**0**=Sunday ）

每个模式都可以是星号 (*****) （表示所有合法值）或逗号分隔的元素列表。元素是上面所示范围中的一个数字，或者由连字符分隔的范围（表示包括上限和下限的范围）中的两个数字。请注意，天的指定可以在两个字段中进行： *monthday* 和 *weekday* 。如果在一个条目中同时指定了这两个字段，则它们是累积的。例如，

0 0 1,15 * 1 *command*

在每月的第一日和第十五日以及每个星期一的午夜运行 *command* 。要仅在一个字段中指定天，请将其他字段设置

为星号 (*) 。例如，

0 0 * * 1 *command*

仅在星期一运行 *command* 。

第六个字段 *command* （crontab 文件中包括空白的行的平衡）是一个由 Shell 在指定时间执行的字符串。将此字段中的百分比符号 (**%**) （除非由反斜杠 (****) 转义）转换为换行符，从而将字段分成“行”。仅命令字段的第一“行”（直到 **%** 或行结束标志）由 Shell 执行。使任何其他“行”可以由该命令用作标准输入。

将忽略空行和其第一个非空字符是 **#** 的行。

cron 使用 POSIX Shell 从用户的 **HOME** 目录 (**/usr/bin/sh**) 调用该命令。它在 **c** 队列中运行（请参阅 *queuedefs*(4)）。

cron 为每个 Shell 提供缺省环境，定义以下内容：

HOME=*user's-home-directory*
LOGNAME=*user's-login-id*
PATH=/usr/bin:/usr/sbin:.
SHELL=/usr/bin/sh

希望执行其 **.profile** 的用户必须在 crontab 条目中或该条目调用的脚本中显式这样做。

如果您的名称出现在文件 **/usr/lib/cron/cron.allow** 中，则您可以执行 **crontab** 。如果该文件不存在，则在您的名称未出现在文件 **/usr/lib/cron/cron.deny** 中时，您可以使用 **crontab** 。仅当 **cron.deny** 存在且为空时，所有用户才可以使用 **crontab** 。如果这两个文件都不存在，则只有 **root** 用户可以使用 **crontab** 。 **allow/deny** 文件在每行中包括一个用户名。

外部语言环境影响

环境变量

LC_CTYPE 用于确定将文件内的文本解释为单字节字符和（或）多字节字符。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或 **LC_MESSAGES** ，或者将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5)）而非 **LANG**。

如果任一国际化变量中包含无效设置，则 **crontab** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。
EDITOR 用于确定在指定 **-e** 选项时要调用的编辑器。缺省编辑器是 vi。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

警告

确保从命令重定向标准输出和标准错误。如果未这样做，则通过邮件将任何生成的标准输出或标准错误发送给用户。

文件

/var/adm/cron	主 cron 目录
/var/adm/cron/cron.allow	允许的用户的列表
/var/adm/cron/cron.deny	拒绝的用户的列表
/var/adm/cron/log	记账信息
/var/spool/cron/crontabs	包含 crontab 文件的目录

另请参阅

sh(1)、 cron(1M)、 queuedefs(4)。

符合的标准

crontab：SVID2、SVID3、XPG2、XPG3、XPG4

名称

crypt - 编码和解码文件

概要

crypt [*password*]

说明

crypt 从标准输入读取文件，并将其输出到标准输出。 *password* 是选择特定转换的密钥。如果未给定任何 *password* ，则 **crypt** 要求从终端输入密钥，并在键入密钥时关闭显示功能。 **crypt** 使用相同的密钥来加密和解密：

```
crypt key <clear >cypher
crypt key <cypher|pr
```

后一种命令将解密文件并输出明文版本。

由 **crypt** 加密的文件与 **ed** 编辑器以加密方式处理（请参阅 *ed*(1) ）的文件相兼容。

加密文件的安全性取决于三种因素：基本加密方法必须难于破解；对密钥空间的直接搜索必须是不可行的；使密钥或明文可变为可见的“潜通路”的功能必须最小化。

crypt 可实现沿用德国 Enigma 生产线设计的单转子计算机，但是其转子具有 256 个元素。对这类计算机的攻击方法已众所周知；因此 **crypt** 提供最弱的安全性。

密钥到计算机的内部设置的转换预先被设计为高开销；即，计算占用了绝大部分时间。但是，如果对密钥进行了限制，例如限制为三个小写字母，则只花费五分钟计算机时间的一小部分便可以读取加密的文件。

由于密钥是 **crypt** 命令的参数，因此密钥对于执行 **ps** 或派生命令（请参阅 *ps*(1) ）的用户可能是可见的。密钥选择和密钥安全性是 **crypt** 最易受到攻击的方面。

举例

以下示例说明了如何使用 **crypt** 来编辑用户要求严格保密的文件：

```
$ crypt <plans >plans.x
key: violet
$ rm plans
...
$ vi -x plans.x
key: violet
...
:wq
$
...
$ crypt <plans.x | pr
key: violet
```

请注意， **-x** 选项是 **vi** 的加密模式，并提示用户输入在加密文件时使用的密钥。

警告

如果输出通过管道传输给 **nroff** ，且在命令行中 未给出加密密钥，则 **crypt** 可能使终端模式处于一种异常状态（请参阅 *nroff*(1) 和 *stty*(1) ）。

如果将使用相同密钥加密的两个或两个以上文件连接起来，并且尝试对连接在一起的文件进行解密，则只有第一个原始文件可以正确被解密。

文件

/dev/tty 用于键入的密钥

另请参阅

ed(1)、 makekey(1)、 stty(1)。

名称

csh - 一种类似于 C 语言语法的 Shell（命令解释程序）

概要

csh [**-cefinstvxTVX**] [*command_file*] [*argument_list* ...]

说明

csh 是一种命令语言解释程序，其中包含命令记录缓冲区、类 C 的语法以及作业控制工具。

命令选项

命令选项的解释如下：

-c 从必需的（单个）后续参数中读取命令。其余所有参数将放在 **argv** 中。

-e 如果有任何调用的命令意外终止或产生非零的退出状态，则 C Shell 将退出。

-f 禁止在主目录中执行 **.cshrc** 文件，从而提高 Shell 的启动速度。

-i 当从设备而非计算机终端（例如另一台计算机）调用时，强制 **csh** 以交互方式进行响应。 **csh** 通常以非交互方式进行响应。如果 **csh** 是从计算机终端调用的，它将始终以交互方式进行响应，而不会考虑所选择的选项。

-n 解析但 不执行命令。这对于检查 Shell 脚本中的语法是有用的。执行所有替换命令（历史记录、命令、别名等）。

-s 从标准输入中获取命令输入。

-t 读取并执行单行输入。

-v 设置 **verbose** Shell 变量，以便在进行历史记录替换后将命令输入回显至标准输出设备。

-x 设置 **echo** Shell 变量，以便在紧接命令执行之前将所有命令回显至标准错误。

-T 禁用 tenex 功能，这些功能使用 ESC 键完成命令/文件名，使用 CTRL-D 列出可用文件（请参阅下面的 CSH 工具部分）

-V 在执行 **.cshrc** 之前设置 **verbose** 变量，以便将所有 **.cshrc** 命令也回显至标准输出。

-X 在执行 **.cshrc** 之前设置 **echo** 变量，以便将所有 **.cshrc** 命令也回显至标准输出。

处理命令选项后，如果参数仍保留在参数列表中，并且未指定 **-c** 、 **-i** 、 **-s** 或 **-t** 选项，则第一个剩余参数将被当作要执行的命令文件的名称。

命令

简单命令由一系列单词组成，第一个单词指定要执行的命令。由竖线 (|) 字符分隔的一系列简单命令构成管道线。管道线中每个命令的输出是管道线中下一个命令的输入。多个管道线系列可以用分号 (**;**) 隔开，以便按顺序执行这些管道线。可以以后台模式执行一系列管道线，方法是：在最后一个条目后添加一个 & 符号。

任何一个管道线都可以放在括号中以构成一个简单命令，而该简单命令又可以成为另一个管道线的一部分。还可以用 || 或 **&&** 将管道线隔开，在 C 语言中，只有当第一个管道线失败或成功时，才能执行第二个管道线。

作业

csh 将 作业与每个管道线相关联，保留一个当前作业的表（由 **jobs** 命令输出），并为这些作业指定小的整数。当使用 **&** 异步启动某项作业时，Shell 将输出类似如下所示的行：

[1] 1234

表示异步启动的作业的作业编号为 1 并具有一个（顶层）进程，其进程 ID 为 1234。

如果用户正在运行某项作业时打算执行其他操作，则可以键入当前定义的 *suspend* 字符（请参阅 *termio*(7)），该字符将向当前作业发送一个停止信号。随后， **csh** 通常会指示作业已被“停止”并输出另一个提示。然后，用户便可以处理此作业的状态，使用 **bg** 命令将其放入后台，运行其他命令，最后再使用前台命令 **fg** 将作业放回前台。 *suspend* 将立即生效，其功能类似于中断，这是因为：在输入该命令时，挂起的输出和未读取的输入将被忽略。存在一个延迟的 *suspend* 字符，在程序尝试 *read*(2) 该字符之前，它不会生成停止信号。如果用户为某项作业准备了一些命令并且希望在作业读取这些命令后停止它们，则提前键入上述字符将非常有用。

当在后台运行的作业试图从终端读取数据时，该作业将停止。通常允许后台作业生成输出，但可以通过 **stty tostop** 命令（请参阅 *stty*(1)）来禁用此功能。如果设置了此 tty 选项，则当后台作业尝试生成输出时便会停止，就像它们在尝试读取输入时会停止一样。在此类系统中，不会将来自终端界面的键盘符号和行挂起符号发送至后台作业。这意味着，后台作业将不受注销或键入中断、退出、挂起和延迟挂起等符号的影响（请参阅 *termio*(7)）。

可以通过多种方法来引用此 Shell 中的作业。 **%** 字符表示作业名称。如果用户希望引用编号为 1 的作业，则可以将其命名为 **%1** 。只需对作业进行命名便可将其转入前台；因此 **%1** 是 **fg %1** 的同义名称，后者将作业 1 转回前台。与此类似，键入 **%1 &** 会将作业 1 转入后台继续运行。还可以使用为启动作业而键入的字符串的前缀（如果这些前缀意义明确）来命名作业；因此，如果只有一个已挂起作业并且其名称以字符串 **ex** 开始，则 **%ex** 通常会重新启动挂起的 *ex*(1) 作业。还可以说是 **%?string** 指定了其文本包含 *string* 的作业（如果只存在一个这样的作业）。

csh 维护当前以及先前作业的表示形式。在有关作业的输出中，当前作业带有 **+** 标记，先前的作业带有 **-** 标记。缩写 **%+** 表示当前作业， **%-** 表示先前的作业。为了与 历史记录机制（下面有介绍）的语法进行紧密类比， **%%** 也是当前作业的同义名称。

csh 在进程改变状态时，会立即获知。只要有作业被阻塞，它通常都会向用户发送通知以便停止所有后续进程，但只是在输出提示之前发送。这种处理方式不会中断用户的工作。但是，如果用户设置了 Shell 变量 **notify** ， **csh** 将立即通知用户后台作业的状态更改。还有一个名称为 **notify** 的 **csh** 内置命令，它会标记单个进程，以便立即报告任意状态更改。缺省情况下， **notify** 标记当前进程。只需在启动后台作业后键入 **notify** ，即可对其进行标记。

如果用户尝试在作业处于停止状态时退出 Shell， **csh** 将发送警告消息： **You have stopped jobs.** 使用 **jobs** 命令进行查看。如果用户执行此操作或立即尝试再次退出， **csh** 不会再次发出警告，并且挂起的作业将被中断（请参阅 *exit*(2)）。

内置命令

内置命令是在 Shell 内部执行而不会衍生新进程的命令。如果内置命令作为管道线的任意部分（最后一个除外）出现，则将在子 Shell 中执行此内置命令。内置命令包括：

alias

alias *name*

alias *name wordlist*

第一种形式输出所有别名。第二种格式输出 *name* 的别名。第三种格式将指定的 *wordlist* 分配为 *name* 的别名。命令和文件名替换在 *wordlist* 上执行。 *name* 不能为 **alias** 或 **unalias** 。

bg [*%job* ...]

将当前（未指定 job ）或指定作业放入后台，以继续这些作业（如果已停止这些作业）。

break 执行命令以在最近的封闭 **foreach** 或 **while end** 之后恢复。将执行当前行中其余的命令。因此可以通过将它们全部写在一行上来实现多级中断。

breaksw

使得 **switch** 中的中断在 **endsw** 后恢复。

case *label***:**

switch 语句中的一个标签，如下所述。

cd

cd *directory_name*

chdir

chdir *directory_name*

将 Shell 的当前工作目录更改为 *directory_name* 。如果未指定， *directory_name* 将缺省为用户的主目录。

如果 *directory_name* 没有作为当前工作目录的子目录进行查找（并且不是以 **/** 、 **./** 或 **../** 开头），则将检查变量 *cdpath* 的所有部分以查看它是否具有子目录 *directory_name* 。最后，如果所有尝试都失败， **csh** 会将 *directory_name* 视为 Shell 变量。如果其值以 **/** 开始，则尝试此操作的目的是查看它是否为目录。另请参阅 *cd*(1) 。

continue

继续执行最近封闭的 **while** 或 **foreach** 。将执行当前行上其余的命令。

default: 在 **switch** 语句中标记缺省情况。缺省应当位于所有其他 **case** 标签后。

dirs 输出目录堆栈；堆栈的顶层位于左侧；堆栈中的第一个目录是当前目录。

echo *wordlist*

echo -n *wordlist*

指定的单词将写入 Shell 的标准输出，单词之间用空格分开。除非指定 **-n** 选项，否则将通过新行中断。另请参阅 *echo*(1) 。

else

end

endif

endsw 请参阅下面对 **foreach** 、 **if** 、 **switch** 和 **while** 语句的描述。

eval *arguments ...*

（与 *sh*(1) 行为相同）。将 *arguments* 作为输入读取到 Shell 和已执行的生成命令。由于将在执行这些替代之前进行解析，因此，这通常用来作为命令结果或变量替代执行生成的命令。

exec *command*

执行指定的命令而不是当前 Shell。

exit

exit (*expression***)**

csh 通过 **status** 变量（第一种形式）的值或指定 *expression* （第二种形式）的值退出。

fg [%*job* ...]

将当前 （未指定*job* ）或指定作业转入前台，以继续这些作业（如果已停止这些作业）。

foreach *name* **(***wordlist***)**

...

end 变量 *name* 被顺序设置为 *wordlist* 的每个成员，并将执行此命令与匹配 **end** 之间的命令序列。（ **foreach** 和 **end** 都必须显示在单独的行中）。

可以使用内置命令 **continue** 继续提前进行循环；使用内置命令 **break** 提前终止循环。如果从终端读取此命令，则将读取一次循环，并在执行循环中的任何语句之前用 **?** 进行提示。如果用户在终端的循环中键入时发生错误，请根据情况使用 erase 或 line-kill 字符进行恢复。

glob *wordlist*

与 **echo** 类似，但不会识别 **** 转义符，单词将以输出中的空字符进行界定。在使用 Shell 对单词列表执行文件名扩展的程序中非常有用。

goto *word*

指定的 *word* 是扩展以生成字符串形式的 **label** 的文件名和命令。Shell 将尽可能多地对其输入进行反绕并搜索前面带有空格或制表符的、形式为 **label:** 的行。在指定的行之后继续执行。

hashstat

输出统计信息行，用以指示内部散列表在定位命令（并避免 **exec** ）时的有效性如何。对于 *path* 的每个部分尝试 **exec** ，此路径中散列函数指示可能的命中，以及各部分中不以 **/** 开始的散列函数。

history [**-h**] [**-r**] [*n*]

显示历史事件列表。如果指定了 *n* ，则仅输出 *n* 个最近的事件。 **-r** 选项将反序输出，即最先输出最近的，而不是最先输出最早的。 **-h** 选项将输出不带前导数字的历史记录列表，以生成适用于 **source** 命令的文件。

if (*expression***)** *command*

如果 *expression* 为真，则将执行包含参数的 *command* 。 *command* 上的变量替换进行得很早，与它对其余部分的 **if** 命令执行替换的时间相同。 *command* 必须为简单命令，而不是管道线、命

令列表、括在括号中的命令列表或别名命令。即使 *expression* 失败，仍将进行输入/输出重定向，也就是说， *command* 不会执行（这是一个缺陷）。

if (*expression1*) **then**

. . .

else if (*expression2*) **then**

. . .

else

. . .

endif 如果 *expression1* 为真，则将执行第一个 **else** 之前的所有命令；否则，如果 *expression2* 为真，则将执行从第一个 **else** 到第二个 **else** 的所有命令，等等。可以包含任意数量的 **else-if** 对，但只需要一个 **endif** 。 **else** 部分也同样为可选项。（单词 **else** 和 **endif** 必须显示在输入行的开始。 **if** 必须单独显示在其输入行中或 **else** 后）。

jobs [**-l**]

列出活动作业。除了列出常见信息外， **-l** 选项还会列出进程 ID。

kill % *job*

kill - *sig* **%** *job* ...

kill *pid*

kill - *sig pid*...

kill -l 将 TERM （终止）信号或指定信号发送到指定的作业或进程。信号按编号或按名称提供（与在去掉 **SIG** 前缀的 **/usr/include/signal.h** 中指定的相同）（请参阅 *signal*(2) ）。信号名称按 **kill -l** 列出。没有缺省值，因此单独使用 **kill** 时不会向当前作业发送信号。如果正在发送的信号为 TERM （终止）或 HUP （挂起），则还会向作业或进程发送一个 CONT （继续）信号。另请参阅 *kill*(1) 。

limit[**-h**][*resource*][*maximum_use*]

限制当前进程以及它所创建的每个进程的使用，使其（单个）不要超过指定的 *resource* 上的 *maximum_use* 。如果未指定 *maximum_use* ，则将显示显示当前限制；如果未指定 *resource* ，则将提供所有限制。

如果指定了 **-h** 标志，则将使用硬性限制而非当前限制。硬性限制在当前限制的值之上又增加了一层限制。只有超级用户可以增加硬性限制，但普通用户可以在允许范围内降低或提高当前限制。

当前可控制的资源包括：

addresspace	进程的最大地址空间（以字节为单位）
coredumpsize	所创建的最大核心转储的大小
cputime	每个进程将要使用的 CPU 的最大秒数

datasize	允许超出程序文本结尾的数据区域的最大增长
descriptors	每个进程所打开文件的最大数量
filesize	可以创建的单个最大文件
memoryuse	进程的驻留集大小可以增长至的最大大小
stacksize	自动扩展的堆栈区域的最大大小

可以将 *maximum_use* 参数指定为其后有比例因子的浮点数或整数： *k* 或 *kilobytes* （1024 字节）、 *m* 或 *megabytes* ，或者 *b* 或 *blocks* （ *ulimit* 系统调用所使用的单位）。对于 *resource* 名称和比例因子，都可以使用名称的明确前缀。可以通过 **csh** 实例来降低 *filesize* ，但只能通过其有效用户 ID 为 *root* 的实例来增加大小。有关详细信息，请参考 *ulimit* 系统调用的文档。

login 终止登录 Shell，将其替换为 **/usr/bin/login** 的实例。这是一种注销方法，目的是与 *sh*(1) 兼容。

logout 终止登录 Shell。如果设置了 *ignoreeof* ，此命令将尤其有用。由于历史原因，还提供了一个类似的函数 **bye** ，该函数用于非登录 Shell 的会话。建议用户不要使用此函数，因为它不是标准 BSD **csh** 的一部分，在将来的版本中可能不受支持。

newgrp 更改调用者的组标识；有关详细信息，请参阅 *newgrp*(1) 。 **newgrp** 会执行新的 Shell，因此当前的 Shell 环境将丢失。

nice
nice +*number*
nice *command*
nice +*number command*

第一种形式将此 Shell 的 *nice* （运行命令优先级）设置为 4（缺省值）。第二种形式将优先级设置为给定的 *number* 。最后两种形式分别以优先级 4 和 *number* 运行 *command* 。具有相应特权的用户可以提高优先级，方法是使用 **nice -***number ... command* 指定负的 niceness，该命令将始终在子 Shell 中执行，并在简单 **if** 语句的命令上添加约束。另请参阅 *nice*(1) 。

nohup [*command*]

如果不使用参数，则可以在 Shell 脚本中使用 **nohup** 对脚本的其余部分忽略挂起。如果使用参数，则可以以忽略挂起的方式运行指定的 *command* 。所有通过 **&** 在后台执行的进程都将有效地被 **nohup** ，如命令部分的“作业”中所述。

notify [*job* ...]

在当前 （未指定*job* ）或指定作业发生变化时， Shell 会异步通知用户；通常先显示通知，然后显示提示。如果设置了 Shell 变量 *notify* ，则此操作将自动发生。

onintr [**-**] [*label*]

控制中断时 Shell 的操作。如果不使用参数， *onintr* 将在发生中断时恢复 Shell 的缺省操作，将终止 Shell 脚本的操作或返回至终端命令输入层。如果指定了 **-** ，将忽略所有中断。如果提供了 *label* ，则当收到中断或当某个子进程因为被中断而终止时， Shell 将执行 **goto** *label* 。

如果在后台允许 Shell 并且忽略中断，则 *onintr* 无效；Shell 和所有调用的命令将继续忽略中断。

popd [**+***n*]

弹出目录堆栈，以返回至新的顶层目录。使用参数将忽略堆栈中的第 *n* 个条目。目录堆栈的元素将从顶部开始以 0 开始计数。由于历史原因，还提供了 **popd** 的同义名称，称为 **rd** 。建议用户不要使用此函数，因为它不是标准 BSD **csh** 的一部分，在将来的版本中可能不受支持。

pushd [*name*] [**+***n*]

如果不使用参数， *pushd* 将交换目录堆栈顶部的两个元素。如果给定 *name* 参数， *pushd* 将更改为新的目录（使用 *cd* ）并将旧的当前工作目录（与在 *csw* 中相同）放入目录堆栈中。如果使用数字参数， *pushd* 将循环目录堆栈的第 *n* 个参数以使其成为顶部元素，并更改至该目录。将以 0 开始从顶部对目录堆栈的成员进行计数。由于历史原因，还提供了 *pushd* 的同义名称，名称为 *gd* 。建议用户不要使用此函数，因为它不是标准 BSD **csh** 的一部分，在将来的版本中可能不受支持。

rehash 使得 *path* 变量中目录内容的内部散列表重新计算。如果在用户登录时有新的命令添加到 *path* 的目录中，则需要此命令。只有当用户向自己的某个目录添加命令或者当系统程序员更改其中一个系统目录的内容时，才需要此命令。

repeat *count command*

将执行 *count* 次指定的 *command* （该命令所受的限制与上面的单行 **if** 语句中的 *command* 所受的限制相同）。只进行一次 I/O 重定向，即使 *count* 为 0 也是这样。

set

set *name*

set *name***=***word*

set *name***[** *index* **]=***word*

set *name***=(** *wordlist* **)**

第一种形式的 **set** 显示所有 Shell 变量的值。其值不是单个词的变量将输出为用括号括起来的单词列表。第二种形式将 *name* 设置为空字符串。第三种形式将 *name* 设置为单个 *word* 。第四种形式将 *name* 的第 *index* 部分设置为 *word* ；此部分必须已经存在。最后一种形式将 *name* 设置为 *wordlist* 中的单词列表。在上述所有情况下，该值均为扩展的命令和文件名。

可以重复这些参数，以在单个 *set* 命令中设置多个值。但请注意，在进行任何设置前，将针对所有参数进行变量扩展。

setenv *name value*

将环境变量 *name* 的值设置为单个字符串 *value* 。最常用的环境变量 **USER** 、 **TERM** 和 **PATH** 将自动导入至 **csh** 变量 *user* 、 *term* 和 *path* ，或从中导出；对于这些变量，没有必要使用 **setenv** 。

shift [*variable*]

如果未指定参数， **argv** 的成员将移到左侧，从而忽略 **argv[1]** 。如果未设置 **argv** 或者为其分配的字符串少于两个，则将发生错误。指定 *variable* 时， *shift* 将在指定的 *variable* 上执行相同的函数。

source [**-h**] *name*

csh 从 *name* 读取命令。 **source** 命令可以嵌套，但如果嵌套太深，Shell 可能耗尽文件描述符或达到最大堆栈大小（请参阅 *maxssiz*(5) ）。如果任何层上的 **source** 发生错误，都将终止所有嵌套的 **source** 命令。通常，不会将执行 **source** 命令期间的输入放入历史记录列表中。可以使用 **-h** 选项将命令放入历史记录列表中而不执行这些命令。

stop [**%***job* ...]

停止正在后台运行的当前（无参数）作业或指定作业。

suspend 使 **csh** 停止，与收到 *suspend* 信号的效果一样。由于 **csh** 通常会忽略 *suspend* 信号，因此这是挂起 Shell 的唯一方法。如果从登录 Shell 中进行尝试，此命令将显示错误消息。

switch (*string***)**

case *str1***:**

. . .

breaksw

. . .

default:

. . .

breaksw

endsw 每个 **case** 标签 (*str1*) 都将针对指定的 *string* 进行连续匹配，该字符串是扩展的第一个命令和文件名。 **case** 标签的形式为模式匹配表示法，以下情况除外：括号表达式中的非匹配列表不受支持（请参阅 *regexp*(5) ）。如果在找到 **default** 标签之前没有匹配的标签，则将从 **default** 标签后开始执行。每个 **case** 标签和 **default** 标签必须显示在一行的开始。 **breaksw** 命令使执行在 *endsw* 之后继续。否则，对 **case** 标签和 **default** 标签的控制将失败，就像在 C 中一样。如果没有匹配的标签并且没有缺省值，则将在 **endsw** 之后继续执行。

time [*command*]

如果没有指定 *command* ，将输出此 Shell 及其子 Shell 所使用的时间汇总信息。如果指定了，则将对简单 *command* 进行计时，并且将输出如 **time** 变量所述的时间汇总信息。当命令完成时，将在必要情况下创建一个额外的 Shell 以输出时间统计信息。

umask [*value*]

显示当前文件创建掩码（未指定 value ）或将其设置为指定的 *value* 。掩码以八进制表示。掩码的通用值包括 **002** ，该值为所有者和组提供了所有权限，为所有其他用户提供了读和执行权限；或 **022** ，该值为所有者提供了所有权限，仅为组和所有其他用户提供了读和执行权限。另请参阅 *umask*(1) 。

unalias *pattern*

将忽略其名称与指定的 *pattern* 相匹配的所有别名。因此， **unalias *** 将删除所有别名。如果 *pattern* 与现有别名不匹配，则不会发生错误。

unhash 禁止使用内部散列表来加速定位所执行的程序。

unset *pattern*

将删除其名称与指定的 *pattern* 相匹配的所有变量。因此， **unset *** 将删除所有变量；此操作具有明显的负作用。如果 *pattern* 没有任何匹配项，则不会发生错误。

unsetenv *pattern*

从环境中删除其名称与指定的 *pattern* 匹配的所有变量。另请参阅上面的 **setenv** 命令和 *printenv*(1) 。

wait 等待所有后台作业终止。如果 Shell 为交互式，则一个中断可以终止等待，此时，Shell 将输出已知未完成的所有作业的名称和作业编号。

while (*expression***)**

. . .

end 尽管指定的 *expression* 的计算结果为非零，仍将计算 **while** 与匹配的 **end** 之间的命令。可以使用 **break** 和 **continue** 来提前终止或继续循环。（ **while** 和 **end** 必须单独显示在它们的输入行中）。如果输入为终端（即非脚本），则首次通过循环时将显示提示，与对待 **foreach** 语句一样。

% *job* 将指定的作业转入前台。

% *job* **&**

继续在后台执行指定的作业。

@

@ *name***=***expression*

@ *name***[***index***]=***expression*

第一种形式将输出所有 Shell 变量的值。第二种形式将指定的 *name* 设置为 *expression* 的值。如果表达式包含 **<** 、 **>** 、 **&** 或 **|** ，则至少这一部分的表达式必须放入括号中。第三种形式将 *expression* 的值分配给 *name* 的第 *index* 个参数。 *name* 及其第 *index* 部分必须已经存在。

***=** 、 **+=** 等运算符与在 C 中一样可用。可以选择使用空格将 *name* 从分配运算符中分开。但是，在分隔 *expression* 各部分时必须使用空格，否则这些部分将成为单个的单词。

特殊后缀 **++** 和 **--** 运算符分别用于增加和减少 *name* （例如 **@ i++** ）。

非内置命令执行

如果要执行的命令不是内置命令， **csh** 将尝试通过 *exec*(2) 执行命令。变量 *path* 中的每一个单词都命名一个目录，Shell 将尝试在该目录中查找命令（如果命令不以 **/** 开始的话）。如果既没有提供 **-c** 又没有提供 **-t** ，Shell 会将这些目录中的名称散列至内部表，以便仅在命令可能驻留在其中的目录中尝试 *exec* 。当搜索路径中存在大量目录时，此命令可以大大加快命令查找速度。如果（通过 **unhash** ）禁用了此机制，或者指定了 **-c** 或 **-t** ，或者 *path*

的任何一个目录的一部分不以 **/** 开始，Shell 会将目录名称与指定的命令名称连接起来以形成文件的路径名，然后 Shell 将尝试执行该文件。

放在括号中的命令将始终在子 Shell 中执行。因此

(cd ; pwd)

将输出 *home* 目录，并在完成时返回至当前目录，但是：

cd ; pwd

在完成时仍将保留在 *home* 目录中。

如果将命令放在括号中，则通常是为了防止 *chdir* 影响当前的 Shell。

如果文件具有执行权限但不是一个可执行的二进制文件，则系统将假定该文件为脚本文件，该脚本文件是当作单独的进程执行的解释程序的数据文件。

csh 将首先尝试加载并执行该脚本文件（请参阅 *exec*(2) ）。如果脚本文件的前两个字符为 **#!** ， *exec*(2) 将期待遵循某个解释程序路径名并尝试作为单独的进程执行指定的解释程序，以读取整个脚本文件。

如果没有命名 **#!** *interpreter* ，并且 Shell 存在 *alias* ，则将在参数列表的开始位置插入 *alias* 的单词，以形成 Shell 命令。 *alias* 的第一个单词应当是要使用的命令的完整路径名。请注意，这是 *alias* 替换的一个特殊的延迟发生案例，它可以在无需修改的情况下将单词插入参数列表。

如果没有命名 **#!** *interpreter* 并且没有 Shell *alias* ，但文件的第一个字符为 **#** ，则将执行由 **$shell** 变量命名的解释程序（请注意，这通常为 **/usr/bin/csh** ，除非用户已重设 **$shell** ）。如果未设置 **$shell** ，则将执行 **/usr/bin/csh** 。

如果未命名 **!#** *interpreter* ，没有 Shell 别名，并且文件的第一个字符不是 **#** ，则将执行 **/usr/bin/sh** 以解释脚本文件。

历史记录替换

历史记录替换使用户可以重复命令，使用来自先前命令的单词作为新命令的一部分，在当前命令中重复先前命令的参数，以及修复先前命令中的拼写或键入错误。

历史记录替换以感叹号 (**!**) 开始。可以在输入流中的任意位置开始替换，但 *cannot* 嵌套。可以在感叹号前面加一个反斜杠，以取消其特殊含义。为了方便起见，当感叹号后面跟有空格、制表符、换行符、等号或左括号时，会将感叹号原样传递至解析程序。任何包含历史记录替换的输入行都将先回显在终端上，然后，系统才会执行该输入行以进行验证。

来自终端、由一个或多个单词组成的命令输入将保存在历史记录列表中。历史记录替换会重新将这些已保存命令的单词序列引入输入流中。先前保存的命令的数量由 **history** 变量控制。无论先前命令的值是什么，都将保存先前的命令。将从 1 开始按顺序对命令进行编号。

用户可以按以下方式引用先前的事件：按事件编号（例如表示事件 10 的 **!10** ）、相对事件位置（例如表示第二个先前事件的 **!-2** ）、完整或部分命令名称（例如表示使用包含初始字符 **d** 的命令的最后一个事件的 **!d** ）以及字符串表达式（例如引用包含字符 **mic** 事件的 **!?mic?** ）。

在不做进一步修改的情况下，这些形式只是重新引入指定事件的单词，单词之间用一个空格隔开。作为一种特

例，**!!** 表示重做；它将引用先前的命令。

要选择命令中的单词，请在事件规范后对所需的单词使用冒号 (**:**) 和标志符。输入行中的单词从零开始编号。基本的单词标志符包括：

0 第一个单词（即命令名称本身）。

n 第 *n* 个单词。

^ 第一个参数。（其作用与 **1** 相同）。

$ 最后一个单词。

a-*b* 从 *a* 到 *b* 单词范围。特殊情况包括 **-***y* 、表示“单词 0 到单词 *y* ”的缩写；和 *x***-** 以及表示从“单词 *x* 到单词 **$** ，但不包括 **$** ”。

***** 从第二个单词到最后一个单词的范围。

% 与搜索序列配合使用，以替换紧靠它前面的匹配单词。

如果参数选择器以 **^** 、 **$** 、 ***** 、 **-** 或 **%** 开始，则可以忽略将命令说明与单词标志符分开的冒号。

单词标志符后面可以跟一系列修饰符，每个修饰符前面带一个冒号。可以定义以下修饰符：

h 通过删除所有后续部分，以仅使用路径名的第一部分。

r 通过删除所有结尾后缀 (.xxx) 来使用根文件名。

e 通过删除根名称来使用文件名的结尾后缀 (**.***xxx*) 。

s */l*/*r*
用 *r* 的值替换指示的命令中的 *l* 值。

t 通过删除所有前导路径名称部分，仅使用路径名称的最终文件名。

& 重复先前的替换。

p 输出新命令但不执行该命令。

q 用引号将替换的单词引起来，以防止进一步替换。

x 与 **q** 相似，但在遇到空格、制表符和换行符时拆分为单词。

g 使用全局 **g** 命令作为另一个修饰符的前缀，以对指定的更改全局进行。命令中的所有单词都将被更改，每个单词一个更改，包括在单引号 (**'**) 或双引号 (**"**) 中的每个字符串均被视为单个单词。

除非前面带有 **g** ，否则将仅对第一个可修改的单词应用修改。如果尝试替换但未能成功完成（例如，请求在仅包含 10 个命令的历史记录缓冲区中替换 **!11** ），将导致错误。

替换的左侧为字符串；而不是 HP-UX 编辑器意义上的正则表达式。可以使用任何字符作为分隔符来代替斜杠 (*/*)。如果在 *l* 或 *r* 字符串中使用分隔符，请使用反斜杠对分隔符进行转义。右侧的 **&** 字符将被左侧的文本代替。**** 还可以对 **&** 进行转义。空 *l* 字符串使用先前的字符串，该字符串来自 *l* ，或来自 **!?***s***?** 中的上下文扫描字符串

s。如果后面紧跟换行符，则可以忽略替换中的结尾分隔符，因为它可能是上下文扫描中的结尾 **?** 。

可以在没有事件规范的情况下指定历史记录引用（如在 **!$** 中一样）。在这种情况下，引用是针对先前的命令而言的，除非先前的历史记录引用发生在同一行上，因为在这种情况下，此形式将重复先前的引用。因此

!?foo?^ !$

提供了来自命令匹配 **?foo?** 的第一个和最后一个参数。

如果输入行的第一个非空白字符是插字符号 (**^**)，则将出现历史记录引用的特殊缩写。这相当于 **!:s^** ，它为在先前行的文本上进行替换提供了方便的快捷方式。因此， **^lb^lib** 可以修复先前命令中 **lib** 的拼写。

最后一点，必要时可以将历史记录替换放入大括号 **{ }** 中，以将其与后面所跟的字符分隔开。因此，接下来

ls -ld ~paul

用户可以执行 **!{l}a** 以实现

ls -ld ~paula

而 **!la** 将寻找以 **la** 开始的命令。

使用单引号和双引号括起来

可以使用以单引号 (**'**) 和双引号 (**"**) 引起来的字符串来阻止全部或部分剩余替换。放在单引号中的字符串可以避免被进一步解释。放在双引号中的字符串仍然可进行变量和命令扩展，如下所述。

在这两种情况下所得到的文本都将成为单个单词（部分或全部）。只有在一种特殊情况下（请参阅下面的命令替换），用双引号引起来的字符串会生成超过一个单词的部分；而用单引号引起来的字符串在任何情况下都不会出现这种情况。

别名替换

csh 维护一个别名列表，用户可以通过 **alias** 和 **unalias** 命令建立、显示和修改这些别名。系统在扫描命令行后会将其解析为明确命令，并将检查每个命令（从左到右）的第一个单词以查看该命令是否具有别名。如果有，将通过可用的历史记录机制重新读取表示该命令的别名的文本，就像该命令是先前的输入行一样。所得到的单词将代替命令和参数列表。如果没有引用历史记录列表，则参数列表将保持不变。

因此，如果 **ls** 的别名为 **ls -l** ，命令 **ls /usr** 将映射至 **ls -l /usr** ，而参数列表将保持不变。同样，如果 **lookup** 的别名为 **grep !^ /etc/passwd** ， **lookup bill** 将映射至 **grep bill /etc/passwd** 。

如果找到别名，则将执行输入文本的文字转换，并且将在经过重新格式化的输入行上重新开始别名创建过程。如果新文本的第一个单词与旧单词一样（通过对其进行标记以防止进一步进行别名创建来实现），则将禁止循环。系统将检测其他循环并且这些循环会导致错误。

请注意，该机制允许别名引入解析程序元语法。因此：

alias print 'pr \!* | lp'

将生成一个命令，该命令使用 *pr*(1) 在行式打印机上输出其参数。

表达式

某些内置命令采用表达式，这些表达式中的运算符类似于 C 中的表达式并具有相同的优先级。这些表达式显示在 **@** 、 **exit** 、 **if** 和 **while** 命令中。可用运算符如下（以优先级递增的方式显示）：

|| && | ^ & == != =~ !~ <= >= < > << >> + - * / % ! ~ ( )

以下列表显示了这些运算符的组合。列表中的优先级从顶部到底部递减：

*** / %**

+ -

<< >>

<= >= < >

== != =~ !~

运算符 **==** 、 **!=** 、 **=~** 和 **!~** 将它们的参数作为字符串进行比较；其他所有运算符则针对数字进行操作。运算符 **=~** 和 **!~** 类似于 **!=** 和 **==** ，不同之处在于：其右侧为 *pattern* （包含 * 、 **?** 和 **[...]** 的实例），左侧的运算符将根据这些值进行匹配。当真正所需的操作只是进行模式匹配时，上述优点可以减少 Shell 脚本中 **switch** 语句的使用。

以 **0** 开始的字符串将被视为八进制数字。空参数或缺失的参数被视为 **0** 。所有表达式的结果为十进制数表示的字符串。请注意，一个表达式中不能有两部分出现在同一个单词中。这些部分前后应当有空格，除非它们与表达式中从语法上对解析程序有意义的一部分相邻： **-** 、 **&** 、 **|** 、 **<** 、 **>** 、 **(** 和 **)** 。

在表达式中也可用，因为原始操作数为包含在大括号 (**{ }**) 中的命令执行，以及格式为 **-***l* 的文件查询，其中 *l* 为下列项之一：

- **r** 读访问
- **w** 写访问
- **x** 执行访问
- **e** 存在
- **o** 所有权
- **z** 零大小
- **f** 纯文本文件
- **d** 目录

对指定的 *filename* 进行命令和文件名扩展并测试，以查看它与实际用户是否具有指定的关系。如果文件不存在或无法访问，则所有查询都将返回假 (0)。如果命令退出时状态为 0，则表示命令执行成功并返回真；否则，命令将失败，并返回假。如果需要更详细的状态信息，则应当在表达式之外执行命令并且检查 **status** 变量。

流控制

csh 包含多个具有下列功能的命令：用户可以使用这些命令来调节命令文件（Shell 脚本）的流控制以及从终端输入中（以受限但有用的方法）来调节流控制。这些命令都是通过强制 Shell 重新读取或跳过部分输入来运行的，并且由于实施原因，这些命令将限制插入某些命令。

foreach 、 **switch** 和 **while** 语句以及 **if** 语句的 **if-then-else** 形式要求主要关键字显示在输入行的单个简单命令中，如下所示。

如果无法查找 Shell 的输入，在无论何时读取循环，Shell 都将缓冲输入并在此内部缓冲器中进行查找，以完成循环所包含的重新读取操作。（就此操作允许的范围而言，后向 **goto** 将在不可查询的输入中继续进行）。

信号处理

csh 通常会忽略 *quit* 信号。以后台模式运行的作业不会受从键盘生成的信号的影响，包括挂起。其他信号则包含 Shell 从其父项中继承的值。可以通过 *onintr* 来控制 **csh** 在 Shell 脚本中的中断和终止信号处理。登录 Shell 将捕获 *terminate* 信号；否则此信号将从 Shell 父项中的状态传递到子项。在任何情况下当登录 Shell 读取文件 **.logout** 时，都不允许出现中断。

命令行解析

csh 在空格和制表符处将输入行拆分为单词。下列例外（解析程序元字符）被视为单独的单词：

&	&
|	竖线;
;	分号;
<	小于号;
>	大于号;
(	左括号;
)	右括号;
&&	双 & 符号;
||	双竖线;
<<	双小于号;
>>	双大于号;
#	注释分隔符

反斜杠 (\) 可以消除这些解析程序元字符的特殊含义。前面带有反斜杠的解析程序元字符将被解释为其 ASCII 值。前面带有反斜杠的换行符 (ASCII 10) 相当于空格。

包含在单引号或双引号中的字符构成单词的一部分。这些字符串中的元字符不会形成单独的单词，包括空格和制表符。在反斜杠或引号对中，前面带有反斜杠的换行符表示真正的换行符。

如果 **csh** 的输入不是终端，则 **#** 字符将引入由换行符中断的注释。

CSH 变量

csh 维护一组变量。每个变量都包含等于零或更多字符串（单词）的值。变量的名称最多由 80 个字母数字组成，并且必须以字母开始。下划线被视为字母。可以通过使用 **set** 和 **unset** 命令来显示和更改变量的值。某些变量为布尔型，也就是说。Shell 不关心这些变量的值是什么，而只关心是否已设置变量。

某些操作以数字方式处理变量。 @ 符号 (**@**) 命令允许执行数字计算，并允许将结果指定给变量。空字符串被视为零，包含多个单词值的任何后续单词都将被忽略。

为输入行创建别名并对其进行解析后以及执行每个命令前，将执行变量扩展，以美元符号 (**$**) 字符为关键字。可以通过在美元符号前面加反斜杠字符 (\) 来阻止变量扩展，包含在双引号 (**"**) 中的变量除外，因为在双引号中将 **always** 进行替换。如果变量包含在单引号中，则任何情况下都不会进行扩展。用单引号引起来的字符串将在后面

进行解释（请参阅命令替换），因此即使要进行变量替换，也会在后面才进行。如果美元符号后面跟有空格、制表符或位于行尾，则该美元符号将按原样传递。

系统将在进行变量扩展前识别输入/输出重定向，并分别进行变量扩展。否则会将命令名称和整个参数列表放在一起进行扩展。

除非包含在双引号中或者指定了 **:q** 修饰词，否则变量替换的结果有可能最终是命令和替换的文件名。在双引号中，其值由多个单词组成的变量将扩展为单个单词的一部分，作为变量值的单词之间用空格分隔。如果在替换时应用了 **:q** 修饰词，变量将扩展为多个单词，单词之间用空格分隔并用引号引起来，以阻止后续命令或文件名替换。

系统提供了以下元序列，用于将变量值引入 Shell 输入中。除非另有说明，否则引用未设置的变量属于错误操作。

$*variable_name*

${*variable_name***}**

解释时，此序列将替换为变量 *variable_name* 值的单词，每个单词之间用空格分隔。大括号将 *variable_name* 与后续字符分隔开，否则该变量将被解释为变量名自身的一部分。

如果 *variable_name* 不是 **csh** 变量，而是在环境中进行设置的，则将使用该值。不能对 非**csh** 变量进行如下所示的修改。

$*variable_name*[*selector*]

${*variable_name*[*selector*]**}**

此修改仅从 *variable_name* 的值中选择某些单词。选择器受变量替换的约束，可以由一个数字或两个数字组成，以短线隔开。变量值的第一个单词的编号为 **1** 。如果范围的第一个数字被省略，则缺省为 **1** 。如果范围的最有一个数字被省略，则缺省为变量 (**$#***variable_name*) 中的总单词数。用作选择器的星号元字符将选择所有单词。

$#*variable_name*

${#*variable_name***}**

该形式提供了变量中的单词数，这对于使用 [*selector*] 选项的形式很有用。

$0 此形式将代替从中读取命令输入的文件的名称。如果文件名未知，将发生错误。

$*number*

${*number***}**

此形式等于变量 **argv** (**$argv[***number***]**) 中的索引选择。

$* 这等同于选择所有 *argv* (**$argv[*]**) 。

可以将修饰词 **:h** 、 **:t** 、 **:r** 、 **:q** 和 **:x** 应用到上面的替换中，正如 **:gh** 、 **:gt** 和 **:gr** 一样。如果命令形式中出现大括号 (**{ }**) ，则修饰词必须显示在括号内。 *The current implementation allows only one* **:** *modifier on each* **$d** *expansion*.

不能使用 **:** 修饰词修改下列替换:

$?*variable_name*

${?*variable_name***}**

如果设置了 *variable_name* ，则替换字符串 **1** ；如果未设置，则替换字符串 **0** 。

$?0 如果已经知道当前输入文件名，则替换 **1** ；如果不知道当前输入文件名，则替换 **0** 。

$$ 替换（父）Shell 的（十进制）进程编号。

$< 替换标准输入中的行，此后不再进行进一步的解释。可以用来以 Shell 脚本的形式从键盘进行读取。

预定义变量和环境变量

以下变量对 Shell 具有特殊含义。在这些变量中， **autologout** 、 **argv** 、 **cwd** 、 **home** 、 **path** 、 **prompt** 、 **shell** 和 **status** 始终由 Shell 设置。除了 **cwd** 和 **status** ，此设置仅在初始化时发生（开始执行 **csh** 时）。除非用户明确修改，否则系统不会修改这些变量。

csh 将 HP-UX 环境变量 USER 复制到 Shell 变量 **user** 中，将环境变量 **TERM** 复制到 **term** 中，将环境变量 **HOME** 复制到 **home** 中，并将 **PATH** 复制到 **path** 中。一旦重置 **csh** 变量， **csh** 便会将这些值复制回环境中。

在窗口环境中，如果 **csh** 检测到窗口大小已发生变化， **csh** 将设置环境变量 **LINES** 和 **COLUMNS** 以匹配新的窗口大小。

argv 此变量将被设置为 **csh** 命令语句的参数。位置参数正是在此变量中进行替换的，也就是说， **$1** 将被 **$argv[1]** 等代替。

cdpath 此变量提供了一个搜索到的备用目录列表，用于在 *chdir* 命令中查找子目录。

cwd 此变量包含当前工作目录的绝对路径名。一旦更改目录（通过使用 *cd* ），系统便会更新此变量。

echo 此变量由 **-x** 命令行选项设置。如果设置了此变量，则所有内置命令及其参数都将在执行前回显至标准输出设备。系统将在替换命令和文件名之前回显内置命令，因为回显之后，这些替换变为可选操作。对于非内置命令，所有扩展都将在回显之前进行。

history 此变量用于创建命令历史记录缓冲并设置其大小。如果未设置此变量，则不会维护命令历史记录而且不会进行历史记录替换。如果 **history** 的值过大，则可能导致 Shell 内存溢出。如果值为 10 或 20，则属于正常状态。所有命令，无论是可执行命令还是不可执行命令，都保存在命令历史记录缓冲区中。

home 此变量包含执行主目录的绝对路径名。变量 **home** 在 HP-UX 环境中进行初始化。波形字符的文件名扩展 (˜) 将引用此变量。

ignoreeof 如果设置了此变量， **csh** 将忽略终端输入设备中的文件结尾字符。 **csh** 在遇到文件结尾情况 （CTRL-**D** 被当作命令行上的第一个字符键入）时，将正常退出。设置 *ignoreeof* 可以防止当前 Shell 被意外 (CTRL-**D**) 终止。但是，为了防止发生 EOF 输入的无限循环， **csh** 在收到 26 个连续的 EOF 时将终止。

mail 此变量包含一个文件列表，其中， **csh** 将检查邮件。命令完成时， **csh** 在生成提示前会定期检查此变量（缺省为 10 分钟）。如果变量包含自上次检查后经过修改（由输入到文件中的邮件生成）的文件名， **csh** 将输出 **You have new mail** 。

如果 **mail** 值的第一个单词为数字，则该数字将指定一个不同的邮件检查间隔（以秒为单位）。

如果指定了多个邮件文件，Shell 将指示 **New mail in** *file_name* ，其中， *file_name* 是包含邮件的文件。

noclobber 此变量将对输出重定向施加限制，以确保文件不会被意外破坏，并确保使用追加重定向 (>>) 的命令可以引用现有文件。

noglob 如果设置了此变量，将禁止文件名扩展。这在 Shell 脚本中非常有用：不处理文件名；或者在获取文件名列表之后并且不需要进行进一步扩展的 Shell 脚本。

nonomatch 如果设置了此变量，则文件名扩展不与任何现有文件匹配的情况将不再是一个错误。如果没有匹配项，则将返回原始模式。对于格式错误的原始模式而言，这仍然是一个错误。例如， **'echo [' ** 仍然会显示错误。

notify 如果设置了此变量， **csh** 将立即（通过用户的标准输出设备）通知用户后台作业已完成。缺省为 **unset** （表示作业刚好在输出提示之前完成）。

path 路径变量的每个单词都会指定一个目录，将在该目录中查找要执行的命令。空单词指定用户当前的工作目录。如果没有 *path* 变量，则只能执行完整路径名称。如果未设置 *path* 并且用户未指定完整路径名称，则 **csh** 将在 **.** （当前目录）和 **/usr/bin** 中搜索命令。如果既没有为 **csh** 指定 **-c** ，也没有为其指定 **-t** 选项，则它通常会在读取 **.cshrc** 后以及每次重置 **path** 变量时，对 **path** 变量中的目录内容进行散列。如果在 Shell 处于活动状态时向这些目录添加了新命令，则有必要执行 **rehash** 以便 **csh** 访问这些新命令。

prompt 此变量允许用户选择自己的提示字符串。将在从交互式终端输入中读取每个命令之前输出提示。如果字符串中显示 **!** ，则将被替换为当前命令历史记录缓冲区事件编号，除非指定了前导 **** 。对于普通用户而言，缺省提示为百分号 (**%**)；对于超级用户而言，缺省提示为 **#** 字符。

savehist 当用户注销时保存在 **~/.history** 中的历史记录列表的行数。如果 **savehist** 的值较大，则会降低 **csh** 的启动速度。

shell 此变量包含 **csh** 程序所驻留的文件的名称。此变量用于 forking Shell 中，以解释设置了执行位但无法由系统执行的文件。（请参阅非内置命令执行的说明）。

status 此变量包含由上一个命令返回的状态值。如果命令意外终止，系统会将 0200 添加到状态变量的值中。意外终止的内置命令将返回退出状态 **1** ，其他所有内置命令的状态将设置为 **0** 。

time 此变量包含控制命令自动计时的数字值。如果设置了此变量，则对于 CPU 占用时间超过指定秒数的所有命令， **csh** 都将输出一行信息到标准输出设备，其中包括用户、系统、实际执行时间加上利用百分比。利用百分比是用户加系统时间与实际时间之比。此信息将在命令完成执行后输出。

verbose 此变量由 **-v** 命令行选项设置。如果设置此变量，则在完成历史记录替换后，将在标准输出设备上输出每个命令的单词。

命令和文件名替换

系统会选择性地将其余的替换、命令和文件名替换应用到内置命令的参数中。这意味着，未评估的表达式部分将不受这些表达式的约束。对于不是 Shell 内部命令的命令而言，将从参数列表中单独替换命令名。此操作将在执行输入-输出重定向之后进行，并且在主 Shell 的子 Shell 中进行。

命令替换

命令替换由包含在着重号 (**'...'**) 中的命令表示。来自此类命令的输出通常在遇到空格、制表符和换行符时被拆分为单个的单词，并将忽略空单词；然后，此文本将替换原始字符串。在双引号中，只有换行符会强制生成新单词；空格和制表符将被保留。

在任何情况下，单个最终换行符都不会强制生成新单词。请注意，这可能会导致命令替换仅生成某个单词的一部分，即使该命令输出完整的行。

文件名替换

每个命令 *word* 都被当作文件名替换（也称为 **globbing** ）的一种模式进行处理，并且被替换为与模式匹配的、经过排序的文件名列表。模式的形式为由 *regexp*(5) 定义的模式匹配表示法，包含以下例外：

- 不支持括号表达式中的非匹配列表。
- 在指定文件名替换的单词列表中，它是一个错误，因为没有与现有文件名匹配的模式，但系统并不要求每个模式都必须匹配。
- 元格式 **a{b,c,d}e** 是“abe ace ade”的快捷方式。系统将保留从左到右的顺序，并在较低的级别对匹配结果进行排序，以维持这种顺序。这种结构可以嵌套。因此：

  ~source/s1/{oldls,ls}.c

  扩展至

  /home/source/s1/oldls.c /home/source/s1/ls.c

  如果 **source** 的主目录为 **/home/source** ，则无论这些文件是否存在，都不会出现错误。同样，

  ../{memo,*box}

  可以扩展至

  ../memo ../box ../mbox

  （请注意，系统不使用匹配 ***box** 的结果对 **memo** 进行排序）。在特殊情况下，系统将原样传递 **{** 、 **}** 和 **{ }** 。

输入/输出

可以使用以下语法对命令的标准输入和标准输出进行重定向:

< *name* 打开文件 *name* （这是扩展的第一个变量、命令和文件名），将其作为标准输入。

<< *word* 读取 Shell 输入，直至遇到与 *word* 相同的行。 *word* 不受变量、文件名或命令替换的约束，在此输入行上进行任何替换之前，系统会先将每个输入行与 *word* 进行比较。除非 **** 、.if n '' 、.if n ''、 **’** 或 **‘** 出现在 *word* 中，否则系统将在插入行上执行变量和命令替换，从而允许 **** 引用 **$** 、 **** 和 **‘** 。除了丢弃最后一个换行符以外，被替换的命令将保留所有空格、制表符和换行符。得到的文本将放在作为标准输入指定给命令的匿名临时文件中。

> *name*
>! *name*
>& *name*
>&! *name* 文件 *name* 用作标准输出。如果文件不存在，则将创建该文件；如果文件存在，则将截断该文件并且其先前的内容都将丢失。

如果设置了 **noclobber** 变量，则该文件不能存在或者必须是一个字符特殊文件（例如终端或 **/dev/null** ）或一个错误结果。这有助于防止意外破坏文件。在这种情况下，可以使用感叹号 (**!**) 形式来禁止此检查。

包含 & 字符 (**&**) 的形式会将标准错误传送至指定的文件和标准输出。系统将以扩展 **<** 输入文件名的方式扩展 *name* 。

>> *name*
>>& *name*
>>! *name*
>>&! *name* 将文件 *name* 用作标准输出，这与 **>** 相同，但会将输出追加至文件结尾。如果设置了变量 **noclobber** ，则对于只有在指定了一种 **!** 形式的情况下才会存在的文件而言，这是一个错误。否则，该变量与 **>** 相似。

命令将接收一种环境，当 Shell 经过输入-输出参数修改后以及该命令出现在管道线中时，系统将在该环境中调用 Shell。因此，与先前的某些 Shell 不同，在缺省情况下，从 Shell 脚本执行的命令无权访问命令文本；而是接收 Shell 的原始标准输入。应当使用 **<<** 机制来表示内置数据。这允许 Shell 脚本作为管道线的一部分发挥功能并允许 Shell 以块方式读取其输入。

可以通过包含标准输出的管道定向对话框输出。只需使用 **|&** 形式，而无需自己使用 **|** 。

CSH 实用程序

完成文件名

将文件名当作参数键入到命令中时，不再需要键入完整名称，而只需键入具有唯一性的缩写即可。如果用户希望系统尝试匹配用户的缩写，请按 ESC 键。然后，系统将为用户完成文件名，并将完整名称回显在用户的终端上。如果缩写与可用文件名不匹配，系统将在终端上发出铃声警告。如果前缀与多个较长的文件名匹配，则可能导致

文件名不完整。在这种情况下，文件名将一直延伸直至出现模糊偏差，并且系统将发出警告。

找到其他目录时，文件名将同样顺利地完成。此外，在此上下文中，可以识别主目录的波形符 (˜) 惯例。

查看文件或目录列表

在键入命令过程中的任何时候，用户都可以询问“哪些文件可用”或“哪些文件与我当前的规范匹配”。因此，当用户键入：

% cd ˜speech/data/bench/fritz/

用户可能希望知道（在 **˜speech/data/bench/fritz** 中）存在哪些文件或子目录，而无需终止正在键入的命令。此时键入 CTRL-**D** 将列出可用的文件。文件以多列格式列出，并按列进行排序。目录和可执行文件分别由结尾 **/** 和 ***** 标识。完成输出后，系统将为用户重新回显命令以便完成。此外，用户可能还希望知道，到目前为止哪些文件匹配前缀，即当前的文件规范。如果用户键入了：

% cd ˜speech/data/bench/fr

系统将输出后面跟有 CTRL-**D** 、在目录 **˜speech/data/bench** 中的前缀为 **fr** 的所有文件和子目录。请注意，上面的示例只是包含空的结尾文件名的这种情况的一个简并案例。（空字符串是所有字符串的前缀）。另外还请注意，对于完成文件名和列表，都需要一个结尾斜杠以传递到新的子目录。请注意简并案例

% ˜^D

将在当前系统上输出登录名的完整列表。

命令名识别

识别和完成命令名的方式与上面识别和完成文件名的方式相同。环境变量 **PATH** 的当前值用于搜索命令。例如

% newa [Escape]

可以扩展至

% newaliases

同时，

% new [Control]-[D]

将列出以 **new** 开始的所有命令（包括 **PATH** ）。如果设置了 Shell 变量 **listpathnum** ，将在 [Control]-[D] 列表的每个命令旁边输出一个指示 **PATH** 中的索引的编号，此操作为可选项。

自动注销

添加了称为 **autologout** 的新 Shell 变量。如果终端在 Shell 顶层保持闲置（无字符输入）的分钟数超过了为 **autologout** 分配的值，系统将自动注销用户。在命令执行时，系统将暂时禁用 **autologout** 功能。 **autologout** 的初始值为 600。如果未设置该值，或将其设为 0，则将完全禁用 **autologout** 。

命令行控制

^R 用于重新输出当前命令行； **^W** 用于清除在当前命令行中输入的最后一个单词。

Sanity

如果用户的终端看起来是从 raw、cbreak 或 noecho 模式中恢复的， C Shell 可以将其恢复至 sane 模式。

保存历史记录缓冲区

csh 具有在登录会话之间保存历史记录列表的功能。如果将 Shell 变量 **savehist** 设置为数字，系统将保存来自历史记录列表的命令事件数。例如，将行

set history=10 savehist=10

放在用户的 **.cshrc** 文件中可以维护长度为 10 的历史记录缓冲区，并在注销时保存整个列表。当用户再次登录时，系统将恢复整个缓冲区。命令将保存在登录目录的 **.history** 文件中。

外部语言环境影响

环境变量

LC_COLLATE 可确定为替换文件名评估模式匹配表示法时使用的排序序列。

LC_CTYPE 确定文本解释为单字节和/或多字节字符，确定字符分类为字符，以及确定模式匹配表示法中按字符类表达式匹配的字符。

LANG 用于确定显示消息的语言。

如果未在环境中指定 **LC_COLLATE** 或 **LC_CTYPE** ，或者将其设置为空字符串，则会将 **LANG** 的值用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则使用缺省的“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **csh** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

警告

.cshrc 文件的构建方式应确保文件不会在标准输出上生成任何输出，也不会生成标准错误，包括在没有关联终端的情况下调用该文件的情况。 *rcp*(1) 会导致追溯 **.cshrc** 的来源，由此文件生成的任何输出（即使是标准错误）都将导致问题。 *stty*(1) 等命令应当放入 **.login** 而非 **.cshrc** 中，以使其输出不会影响 *rcp*(1) 。

csh 具有某些限制条件。单词或环境变量的长度不能超过 10240 个字符。系统将参数列表限制为 10240 个字符。包含文件名扩展的命令的参数数目被限制为参数列表中允许的字符数的六分之一。命令替换只能替换参数列表中允许的字符数。

为了检测循环，Shell 将单个行上的 **alias** 替换数量限制为 20。

从停止处重新启动命令时，如果 **csh** 从中启动的目录不同于当前目录，则 输出它的启动目录；这可能会出现误导（即错误），因为作业可能已在内部更改了目录。

不能停止/重新启动 Shell 内置功能。如果尝试停止，系统还将无法正常处理 **a ; b ; c** 形式的命令序列。如果中断 **b** ，Shell 将立即执行 **c** 。如果此扩展源自 **alias** ，则这一点尤其值得注意。它可以将命令序列放入括号中，以强制将其放入子 Shell 即 **(a ; b ; c)** 中。

由于 csh 要求信号处理，系统将在执行命令前禁用中断并在命令开始执行时恢复中断。在指定命令与系统识别到中断之间，可能存在几秒钟的延迟。

在启动进程后对 tty 输出进行控制属于原始操作；也许这可以使得某用户在一个好的虚拟终端接口上工作。在虚拟终端接口上，使用输出控制可以执行更多更有趣的工作。

别名替换通常用来简单地模拟 Shell 过程；应当提供 Shell 过程，而非别名。

系统不会将循环中由 **?** 提示的命令放入 *history* 列表。应当解析控制结构，而不是将其当作内置命令识别。这样可以将控制命令放在任意位置，将控制命令与 **|** 合并，以及将控制命令与 **&** 和 **;** 元语法配合使用。

应当可以在命令替换的输出上使用 **:** 修饰词。应当允许在 **$** 替换上使用所有或多个 **:** 修饰词。

仅在用户首次尝试识别时检查终端类型。

要在系统中与 **PATH** 一起列出所有命令，请输入 **[Space]-[Ctrl]-[D]** 。

csh 元序列 **!~** 不起作用。

在国际环境中，字符排序由 **LC_COLLATE** 的设置确定，而不是由计算机排序序列中的字符值的二进制排序确定。这会给它带来某种伴随而来的危险，尤其当在文件名生成模式中使用范围表达式时。例如，命令，

rm [a-z]*

可能期望匹配所有以小写字母字符开头的文件名。但是，如果字母排序由 **LC_COLLATE** 指定，则它还将匹配以大写字符开头的文件名（以及以重音字母开头的文件名）。相反，在挪威语等语言中，它将无法匹配排在 **z** 后面的字母。

在国际环境中匹配特定字符类的正确（而安全）的方式是使用以下形式的模式：

rm [[:lower:]]*

它使用 **LC_CTYPE** 来确定字符类并针对所有支持的语言和代码集以可预测的方式工作。对于在非国际化系统上生成的 Shell 脚本（或不考虑上述危险），建议在非 NLS 环境中执行这些脚本。这要求将 **LANG** 、 **LC_COLLATE** 等设置为“C”或完全不进行设置。

csh 通过在自身与命令之间创建一个管道来实施命令替换。如果根文件系统已满，则替换的命令将无法写入管道。因此，Shell 不会从命令中收到任何输入，并且替换的结果为空。尤其是，在此类情况下使用变量指定的命令替换将导致系统在无提示的情况下为变量分配一个空值。

在一个通过符号链接的目录中进行相对路径更改（例如 **cd ..** ），将导致 **csh** 对工作目录的了解与符号路径而不是物理路径放在一起。

在 HP-UX 9.0 之前的版本中，当从文件中获取其输入时，如果 **csh** 无法执行命令（例如无法查找命令），它将立即退出。从 9.0 开始， **csh** 开始继续尝试执行文件中的其余命令。但是，如果出于兼容性目的而需要执行旧操作，请将环境变量 **EXITONERR** 设置为 1。

作者

csh 由加州大学伯克利分校和 HP 联合开发。

文件

~/.cshrc 在开始执行时由每个 Shell 溯源（执行）的 csh 脚本。请参阅警告

~/.login 在登录时，由登录 Shell 在 **.cshrc** 之后溯源（执行）的 csh 脚本。

~/.logout 由登录 Shell 在注销时溯源（执行）的 csh 脚本。

/etc/passwd **~name** 的主目录来源。

/usr/bin/sh 标准 Shell，用于不是以 **#** 开头的 Shell 脚本。

/etc/csh.login 启动 csh 登录时在 **~/.cshrc** 和 **~/.login** 之前溯源（执行）的 csh 脚本（与 POSIX Shell 中的 **/etc/profile** 类似）。

/tmp/sh* **<<** 的临时文件。

另请参阅

cd(1)、 echo(1)、 kill(1)、 nice(1)、 sh(1)、 umask(1)、 access(2)、exec(2)、 fork(2)、 pipe(2)、 umask(2)、 wait(2)、 tty(7)、 a.out(4)、 environ(5)、 lang(5)、 regexp(5)。

《Shells Users Guide》中的 C Shell 教程。

名称

csplit - 文件内容拆分

概要

csplit [**-s**] [**-k**] [**-f** *prefix*] [**-n** *number*] *file arg1* ... [*argn*]

说明

csplit 可读取 *file* ，将它拆分为 *arg1* ... *argn* 定义的 *n*+1 个部分，并将结果放在单独的文件中。除非使用了 **-n** *number* 选项允许更多的输出文件名，否则允许的最大参数数目（从 arg1 到 *argn* ）是 99。如果指定了 **-f** *prefix* 选项，则得到的文件名是 *prefix***00** 到 *prefixNN* ，其中 *NN* 是 *n* 的两位数值，当 *n* 小于 10 时将使用前导零。如果未指定 **-f** *prefix* 选项，则使用缺省文件名 **xx00** 到 **xx***NN* 。 *file* 的拆分方式如下：

缺省 文件名	加前缀的 文件名	内容
xx00	前缀 **00**	从文件开头直到（但不包括） *arg1* 引用的行。
xx01	前缀 **01**	从 *arg1* 引用的行到 *arg2* 引用的行。
		.
		.
		.
xx*NN*	前缀 *NN*	从 *argn* 引用的行到文件的末尾。

如果 *file* 参数是 **-** ，则使用标准输入。

csplit 支持基本正则表达式语法（请参阅 *regexp*(5) ）。

选项

csplit 可识别下列选项：

-s 不输出所有字符计数（ csplit 通常为创建的每个文件输出字符计数）。

-k 将以前创建的文件保留原样（如果出现错误，则 csplit 通常会删除创建的文件）。

-f *prefix* 将创建的文件命名为 *prefix***00** 到 *prefixNN* （缺省值为 **xx00** 到 **xx***NN* ）。

-n *number* 输出文件名后缀将使用 *number* 指定的位数，而不是缺省值 **2** 。这允许创建超过 100 个的输出文件。

csplit 的参数（ arg1 到 *argn* ）可以是下列各项的任意组合：

/regexp/ 创建一个文件，其中包含从当前行直到（但不包括）与 *regexp* 匹配的行的部分。新的当前行将变为与 *regexp* 匹配的行。

*/regexp/***+***n*
*/regexp/***-***n* 创建一个文件，其中包含的部分为从当前行直到（但不包括）与 *regexp* 匹配的行之前 (**-***n*) 或之后 (**+***n*) 的第 *n* 行（例如， **/Page/-5** ）。新的当前行将变为与 *regexp*±*n*

加减 *n* } 行匹配的行。

%regexp% 基本等效于 */regexp/* ，但不为此部分创建任何文件。

line_number 创建一个包含从当前行直到（但不包括） *line_number* 的文件。新的当前行将变为 *line_number* 。

{*num*} 重复参数。此参数可以跟在上述任一参数形式的后面。如果它跟在 *regexp* 参数后面，则该参数将会再被应用 *num* 指定的次数。如果它跟在 *line_number* 后面，则从该位置开始按 *line_number* 指定的行数将文件拆分 *num* 次，直到到达文件结尾或超过 *num* 次数。

{*} 根据需要多次重复以前的操作数以完成输入。

将包含对 Shell 有意义的空白或其他字符的所有 *regexp* 参数括在适当的括号中。正则表达式不得包含嵌入的换行符。 **csplit** 既不会更改也不会删除原始文件；根据需要删除该文件是用户的工作。

外部语言环境影响

环境变量

LC_COLLATE 可确定在计算正则表达式时使用的排序序列。

LC_CTYPE 可确定与正则表达式中字符类表达式匹配的字符。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_COLLATE** 、 **LC_CTYPE** 或 **LC_MESSAGES** ，或者将其设置为空字符串，则会将 **LANG** 的值用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则使用缺省的 “C” （请参阅 *lang*(5) ） 而非 **LANG**。如果任一国际化变量包含无效设置，则 **csplit** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

消息是自述性的，但以下消息例外：

arg - out of range

该消息表示给定的参数引用的不是位于当前位置和文件末尾之间的行。如果在重复计数结束之前文件已结束，也会出现此警告。

举例

创建四个文件，从 **cobol00** 到 **cobol03** 。在编辑“拆分”后的各文件后，将它们重新组合为原始文件，删除其原始内容。

csplit -f cobol file ’/procedure division/’ /par5./ /par16./

执行编辑操作

cat cobol0[0-3] > file

每 100 行拆分一次文件，该文件最多包含 10,000 行（100 个文件）。如果存在的行少于 10,000 行，则 **-k** 选项将使创建的文件保留（仍将显示错误消息）。

csplit -k file 100 '{99}'

假定 **prog.c** 遵循常规的 C 编码约定（在行首用 } 终止例行程序），创建一个包含 **prog.c** 中每个单独 C 例程（最多 21 个）的文件。

csplit -k prog.c '%main(%' '/^}/+1' '{20}'

另请参阅

sh(1)、 split(1)、 environ(5)、 lang(5)、 regexp(5)。

符合的标准

csplit：SVID2、SVID3、XPG2、XPG3、XPG4

名称

ct - 为远程终端启动 getty（呼叫终端）

概要

ct [-w *n*] **[-x** *n*] **[-h] [-v] [-s** *speed*] *telno...*

说明

ct 拨通 *telno* （即连接到终端的调制解调器的电话号码），然后为该终端启动 *getty*(1M) 进程。

ct 尝试文件 **/etc/uucp/Devices** 中列出的每条线路，直到找到具有适当属性的可用线路，或直到运行完文件中的所有条目。如果没有空闲的线路，则 **ct** 会询问是否需要等待线路，如果需要等待，应在放弃前等待多长时间。 **ct** 将以一分钟为间隔再次搜索可用的线路，直到超出指定的限制。请注意，在通常情况下， **ct** 不连接当前的 tty 线路，所以该线路可以对呼入做出应答。这是因为 **ct** 假设当前的 tty 线路已连接到终端，以便启动 **getty** 进程。

telno 参数可指定由字符 **0** 到 **9** 、 **-** 、 **=** 、 ***** 和 **#** 组成的电话号码。使用等号表示二次拨号音，使用减号表示在适当的位置延迟。 *telno* 最长为 31 个字符。如果已指定多个电话号码，则 **ct** 将接连尝试每个电话号码直到其中一个做出应答；这对于指定备用拨号路径来说是非常有用的。

如果 **ct** 没有连接当前线路，而 **ct** 又准备使用相同的线路再次连接时，则不应在该线路上启动 **getty** 。要启动的话，请将 **inittab** 文件中该线路的条目设置为 **uugetty** 而不是 **getty** （请参阅 *inittab*(4) ）。

选项

ct 可识别下列选项和命令行参数：

-w*n* 如果线路忙碌，则指示 **ct** 等待线路，等待的最长时间为 *n* 分钟。如果已指定该选项，则 **ct** 不会询问用户是否要等待线路。

-x*n* 在标准错误输出中生成程序执行情况的详细输出。该选项用于调试。调试级别 *n* 是单个数字；最有用的值是 **-x9** 。

-h 禁止 **ct** 断开（“挂断”）当前的 tty 线路。如果用户所使用的 tty 线路与 **ct** 用来启动 **getty** 的线路不同，则该选项是非常必要的。

-v 详细模式。 **-v** 选项与 **-h** 选项一起使用，它可使 **ct** 将正在运行的叙述性信息发送到标准错误输出流中。

-s*speed* 设置数据速率，其中 *speed* 以波特表示。缺省速率为 **1200** 。

当目标终端上的用户注销后， **ct** 会提示 **Reconnect?** 。如果响应以字母 **n** 开头，则线路会断开。否则，会重新启动 **getty** ，并输出 **login:** 提示。

当然，目标终端必须连接到可以自动响应呼入的调制解调器上。

文件

/var/adm/ctlog
/etc/uucp/Devices

另请参阅

cu(1)、 login(1)、 uucp(1)、 getty(1M)、 uugetty(1M)。

名称

ctags - 创建标记文件

概要

ctags [-xvFBatwu] *files* ...

说明

ctags 根据指定的 C、Pascal 和 FORTRAN 源为 *ex*(1) （或 *vi*(1) ）创建标记文件。 **tags** 文件在一组文件中提供指定对象（对于 C 包括函数、具有参数的宏和类型定义；对于 Pascal 包括过程、程序和函数；对于 FORTRAN，包括子例行程序、程序和函数）的位置。标记文件的每一行都包含对象名、定义它的文件和对象定义的地址规范。输出结果按升序进行排序（请参阅下面的“环境变量”）。除 C 语言 *typedefs* 外的所有对象都是根据一种模式，即带行号的 *typedefs* 进行搜索的。说明符在行上的单独字段中提供，由空格或制表符分隔。使用 *tags* 文件， **ex** 可以快速找到这些对象的定义。

-x 使 **ctags** 输出简单的函数索引。其实现方法是汇集一个列表，包含函数名、定义每个函数的文件名、出现每个函数名的行号以及每行中的文本。然后在标准输出中输出该列表。不创建或更改任何 *tags* 文件。

-v 在标准输出上生成页索引。此列表包含函数名、文件名和该文件内的页码（假定含有 56 行的页与 *pr*(1) 匹配）。

名称以 **.c** 或 **.h** 结尾的文件被视为 C 源文件，并在其中搜索 C 例行程序和宏定义。对于其他文件，首先检查它们是否包含任何 Pascal 或 FORTRAN 例行程序定义；如果不包含，则再次对其进行处理以查找 C 定义。

其他选项如下:

-F 使用向前搜索模式 (**/**...**/**) （缺省值）。

-B 使用向后搜索模式 (**?**...**?**) 。

-a 将文件中的信息添加到 *tags* 文件。与根据原始文件重新建立 *tags* 文件不同，这可能会导致在 *tags* 文件中同一符号被输入两次。应该小心使用此选项，且仅在非常特殊的情况下使用。

-t 为类型定义创建标记。

-w 不输出警告诊断消息。

-u 更新 *tags* 中的指定文件；即，删除对这些文件的所有引用，并向文件中添加新值，如上面的 **-a** 选项那样。（请注意，此选项的实现相当慢；通常在仅重新建立 *tags* 文件时稍快一些)

在 C 程序中会对标记 **main** 进行特殊处理。形成的标记是通过将 **M** 添加到文件名的开头，删除任何尾部的 **.c** ，并删除前导路径名部分而创建的。这样，就可以在具有多个程序的目录中使用 **ctags** 。

外部语言环境影响

环境变量

LC_COLLATE 用于确定对输出排序的顺序。

LC_CTYPE 用于确定如何解释注释和字符串内的单字节字符和（或）多字节字符。

如果未在环境中指定 **LC_COLLATE** 或 **LC_CTYPE** ，或者将其设置为空字符串，则会将 **LANG** 的值用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **ctags** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集，但不支持多字节字符文件名。

诊断信息

Too many entries to sort.

获取附加堆空间的尝试失败；无法执行排序。

Unexpected end of function in file *file*, **line** *line*.

标记文件可能不正确。

在第一列中意外发现了 **}** 字符。这可能会导致 *tags* 文件中的条目不正确。

Duplicate entry in file *file*, **line** *line*: **name. Second entry ignored.**

在同一文件中两次检测到相同名称。仅为找到的第一个名称建立了一个 *tags* 条目。

Duplicate entry in files *file1* **and** *file2*: *name* **(Warning only).**

在两个不同的文件中检测到相同的名称。仅为找到的第一个名称建立了一个 *tags* 条目。

举例

在当前目录中为其中的所有 C 源文件 (***.c**) 和头文件 (***.h**) 创建名为 **tags** 的标记文件：

ctags *.[ch]

当标记文件存在于当前目录中时，就可以将其用于支持标记文件的命令（如 **vi** ，请参阅 *vi*(1) ）。

将标记文件用于 **vi** 以编辑以下源文件之一中的特定函数 **myfunc()** ：

vi -t myfunc

在使用 **vi** 编辑 C 源文件时，使用 **ex**-mode 标记命令编辑函数 **myfunc()** ：

:tag myfunc

使用 **vi** 在 **myprog.c** 文件中查找 **main()** ：

vi -t Mmyprog

在使用 **vi** 时，在文件 **myprog.c** 中查找 **main()** （不必是当前正在编辑的文件）：

:tag Mmyprog

警告

FORTRAN 和 Pascal 的 **functions** 、 **subroutines** 和 **procedures** 的识别方式非常简单。不尝试处理块结构；如果在不同块中存在同名的两个 Pascal 过程，则将生成警告消息。

确定是查找 C 还是 Pascal 和 FORTRAN 函数的方法是近似法，而且可能被异常程序所欺骗。

ctags 不能识别 **#ifdefs** 和 Pascal 类型。

它依赖于格式正确的输入来检测 *typedefs* 。

使用 **-tx** 时仅显示类型定义的最后一行。

ex 不能处理含有若干相同标记的 *tags* 文件；它仅选择其（非线性）搜索找到的具有该标记的第一个条目。可以使用 **-u** 或 **-a** 选项或者通过编辑 *tags* 文件来创建这种文件。

如果多个（函数）定义出现在一个行上，则只将第一个定义编入索引。

作者

ctags 由加州大学伯克利分校开发。

文件

tags	输出标记文件
OTAGS	由 **-u** 使用的临时文件

另请参阅

ex(1)、 vi(1)。

符合的标准

ctags： XPG4

名称

cu - 调用另一个 (UNIX) 系统；仿真终端

概要

cu [**-s** *speed*] [**-l** *line*] [**-h**] [**-q**] [**-t**] [**-d** *level*] [**-e**|**-o**] [**-m**] [**-n**] [*telno* | *systemname*| **dir**]

XPG4 语法:

cu [**-s** *speed*] [**-l** *line*] [**-h**] [**-q**] [**-t**] [**-d**] [**-e**|**-o**] [**-m**] [**-n**] [*telno* | *systemname*| **dir**]

说明

cu 可调用另一个系统，该系统通常是 UNIX 操作系统，但也可以是终端或非 UNIX 操作系统。 **cu** 管理系统间的所有交互操作，包括可能的 ASCII 文件传输。

选项

cu 可识别下列选项及命令行参数:

-s*speed* 指定传输速率（110、 150、 300、 600、 1200、 2400、 3600、 4800、 7200、 9600、 19200）。缺省值为 300。

-l*line* 指定将用作通信线路的设备名。该选项可用于覆盖具有所需速率的第一个可用线路的搜索。当指定了 **-l** 选项而未指定 **-s** 选项时，将从文件 **/etc/uucp/Devices** 中获得线路的速率。当同时使用了 **-l** 和 **-s** 选项时， **cu** 将搜索 **/etc/uucp/Devices** 以确定所需线路的请求速率是否可用。如果可用，则以请求的速率建立连接；否则，将输出错误消息且不进行调用。指定的设备通常是直接连接的异步线路（例如 **/dev/ttyapb** ）。这时，不需要提供电话号码，但可以使用 **dir** 字符串指出无需拨号程序。如果指定设备与自动拨号程序相关联，则必须提供电话号码。

-h 模拟本地回应，支持对其他期望将终端设置为半双工模式的计算机系统的调用。

-q 使用 ENQ/ACK 握手机制（远程系统发送 ENQ， **cu** 发送 ACK。）

-t 拨打设置为自动应答的 ASCII 终端时使用。设置了回车与回车换行对的相应映射。

-d*level* 输出诊断跟踪消息。 *level* 为从 0 到 9 的数字， *level* 越高，生成的诊断消息越详细。

-d （仅适用于 XPG4。）输出诊断跟踪消息。诊断级别始终为 9。

-e（**-o**） 生成发送至远程的数据的奇（偶）效验。

-m 指定有调制解调器控制的直接线路。 **cu** 忽略调制解调器的控制。

-n 使 **cu** 所拨的电话号码通过用户交互式请求而不是从命令行获得。

telno 使用自动拨号程序时，对于 *telno* ，等号表示备用拨号音，负号表示适当放置在 *telno* 字符串中的延迟。

systemname 可以使用 UUCP 系统名而不是电话号码（请参阅 *uucp*(1) ）；这时， **cu** 从 **/etc/uucp/Systems** 中获取相应的直接线路或电话号码（包括相应的波特率）。 **cu** 将拨打 **Systems** 文件中 *systemname* 的每个电话号码或直接线路，直到建立了连接，或尝试

了所有条目为止。

dir 使用 **dir** 可确保 **cu** 使用 **-l** 选项指定的线路。

建立连接后， **cu** 以两个进程运行：

- 传输进程从标准输入读取数据，并且除了以 ~ 开头的行外，将其他行传送至远程系统；
- 接收进程接收远程系统的数据，并且除了以 ~ 开头的行之外，将其他行传送至标准输出。

通常，使用自动DC3/DC1协议控制远程输入，以确保缓冲区不会溢出。“提示握手”可用于控制将 ASCII 文件传输至不允许预先键入、而要求仅在给出提示后才发送数据的系统。下文有详细叙述。以 ~ 开头的行具有特殊意义。

传输进程命令

传输进程可解释下列命令：

~., **~..** 终止会话。在硬连接 (hard-wired) 线路上， **~.** 会发送几个 EOF 字符来退出会话，而 **~..** 不发送 EOF 序列。一般说来，使用 **~..** 时，远程硬连接的计算机不能察觉到连接的断开。对于拨号连接， **~.** 和 **~..** 没有区别。

~! 退出到本地系统的互动式 Shell。

~!*cmd* ... 在本地系统上运行 *cmd* （通过 **sh -c** ）。

~& 类似于 **~!** 但会终止接收进程，直到从 Shell 返回后重启该进程。该选项可用于调用从通信线路（接收进程将竞争输入）读取数据的子进程。

~&*cmd* ... 在本地系统中运行 *cmd* （通过 **sh -c** ），并终止接收进程，以后重启。

~| *cmd* 在本地系统上用管道将来自远程系统的数据通过标准输入传送到 *cmd* 。若要终止，请使用 **~&** 或 ~| 命令重置。

~| 重置位于 ~|**cmd** 命令后的接收进程。

~$*cmd* ... 在本地运行 *cmd* 并将其输出发送至远程系统。

~%cd 更改本地系统上的目录。 注意： **~!cd** 将使该命令由子 Shell 运行，完成时返回到当前目录。

~%take *remote_source_file* [*local_destination_file*]

将 *remote_source_file* 文件从远程系统复制到本地系统的 *local_destination_file* 文件中。如果未指定 *local_destination_file* ，则都使用 *remote_source_file* 参数。

~%put *local_source_file* [*remote_destination_file*]

将 *local_source_file* 文件从本地系统复制到远程系统的 *remote_destination_file* 文件中。如果未指定 *remote_destination_file* ，则都使用 *local_source_file* 参数。

~~ ... 将 ~ ... 行发送至远程系统。如果在远程系统中使用 **cu** 来访问第三个远程系统，请发送 **~~.** 使第二个远程 **cu** 退出。

~%break 发送一个中断信号至远程系统。

~%nostop 在 DC3/DC1 输入控制协议和无输入控制之间切换。如果远程系统无法正确响应 DC3 及 DC1 字符，此选项很有用。

~%< *file* 使用 提示握手功能将本地文件的内容发送至远程系统。一次只读取指定文件的一行，当收到 提示序列时，每一行会被发送至远程系统。如果在发生 提示超时时仍未收到任何提示，仍然会发送该行。如果将超时设置为 0 秒，或者如果提示序列的第一个字符为空字符 (^@)，则无论远程系统是否生成提示，都似乎立即发生了握手过程。该功能主要用于促成将 ASCI 文件从 HP-UX 传输到运行 MPE 的 HP 3000 系统。其实现方法通常是运行 MPE **FCOPY** 实用程序，并给出 **from=;to=***destfile***;new** 命令，然后运行 **cu** 输入转移以将文件发送至 **FCOPY** （将该文件保存在 *destfile* 中）。该功能在其他系统中可能也很有用，例如运行 RTE 的 HP 1000 系统。

~%setpt *n* 放弃前等待提示的秒数。缺省值为 2 秒。指定 0 秒的超时会禁用握手；即，握手似乎瞬间完成。

~%setps *xy* 将握手提示设置为字符 *xy* 。缺省为 DC1。提示可以是任何一个或两个字符。要指定 *x* 或 *y* 的控制字符，可使用 Ctrl-X 形式，即在字符前加一个插字符号，如 **^X** 。可用 ^@ 指定空字符。（提示中的第一个空字符表示“空”提示，看上去总能实现握手）可用 ^^ 指定插字符号。

~%>[**>**]*file* 使输出从远程系统转移至指定的文件，直到给出了另一个 **~%>** 命令。当一个输出转移处于活动状态时，键入 **~%>** 会终止该转移，而 **~%>** *anotherfile* 将终止它并开始一个新转移。输出转移通过 **~&** 子 Shell 仍然保持激活状态，但是如果输入/输出转移与 **~%take** 或 **~%put** 混杂在一起，可能会发生不可预知的结果。 **~%>>** 命令可将数据追加到指定文件。请注意，由 发送进程解释的这些命令与下面介绍的 **~>** 命令无关，它们由接收进程解释。

~*susp* 挂起 **cu** 会话。 *susp* 是调用了 **cu** 时，在终端上设置的挂起字符（通常为 **^Z** — 请参阅 *stty*(1) ）。与其他所有以波形字符开头的行一样，必须通过按 **Return** 键来终止 ~*susp* 行。

接收进程

通常， 接收进程将数据从远程系统复制到其标准输出中。以 **~>** 开头的远程数据行会启动将输出转移到一个文件中。完整序列为:

~>[**>**]**:** *file*
将零行或多行写入至文件中
~>

远程数据将被转移到（如果使用了 **>>** ，则被追加）到 *file* 。末尾的 **~>** 将终止转移。

在远程系统上，若要使用 **~%put** 需要 *stty*(1) 和 *cat*(1) 。还要求远程系统上的当前 erase 和 kill 字符与本地系统中的当前字符相同。会在适当位置插入反斜杠。

要使用 **~%take** ，要求远程系统支持 **echo** 和 **cat** 命令（请参阅 *echo*(1) 和 *cat*(1) 。此外，如果在不进行扩展的情况下复制制表符，应该在远程系统上设置 **stty tabs** 模式。在连接至使用第 8 位作为校验位的计算机时，应该在本地系统中设置 **stty istrip** 模式。

当在 X 系统上使用 **cu** 连接至 Y 系统，随后在 Y 系统上使用该命令连接至 Z 系统时，如果使用了 **~~** ，则可执行 Y 系统上的命令。例如，在 X 系统上使用键盘时，可在 Z、X 和 Y 系统上执行 **uname** ，如下所示，其中 1、3、5 行为键盘命令， 2、4、6 行为系统响应：

uname
Z
~!uname
X
~~!uname
Y

综上所述， ~ 可使命令在初始计算机上执行； ~~ 使命令在计算机链中的下一个计算机上执行。

外部语言环境影响

环境变量

LANG 用于确定显示消息的语言。

如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省的“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量中包含无效设置，则 **cu** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

正常退出时退出代码为 0；否则为非零值（各种值）。

举例

使用 1200 波特率拨打号码为 9 201 555 1212 的系统：

cu -s1200 9=2015551212

如果未指定速率，则缺省为 300。

登录通过直接线路连接的系统：

cu -l/dev/tty*XpX* **dir**

拨打具有指定线路和速率的系统：

cu -s1200 -l/dev/tty*XpX* **dir**

使用指定线路拨打系统：

cu -l/dev/cul*XpX* **2015551212**

使用系统名 (**yyyzzz**)：

cu yyyzzz

直接连至调制解调器:

cu -l/dev/culXX -m dir cu -l/dev/cu1XX -m dir

警告

cu 在内部缓冲输入。

作者

cu 由 AT&T 和 HP 开发。

文件

/etc/uucp/Systems
/etc/uucp/Devices
/etc/uucp/Dialers
/var/spool/locks/LCK..(tty-device)
/dev/null

另请参阅

cat(1)、 ct(1)、 echo(1)、 stty(1)、 uname(1)、 uucp(1)、 uuname(1)。

符合的标准

cu：SVID2、SVID3、XPG2、XPG3、XPG4

名称

cut - 从文件的每一行提取所选字段

概要

cut -c *list* [*file*]...

cut -b *list* [**-n**] [*file*]...

cut -f *list* [**-d** *char*] [**-s**] [*file*]...

说明

cut 从表中提取列或从文件中各行提取字段；通过数据库用语来映射某个关系。 *list* 所指定的字段可以具有固定的长度（在使用 **-c** 或 **-b** 选项时根据字符或字节在行中的位置来定义），或者，各个行可以具有不同的长度，并用制表符（当使用 **-f** 选项时）等字段定界符来标记。 **cut** 可用作过滤器；如果未给定文件，则使用标准输入。

处理单字节字符集时， **-c** 和 **-b** 选项是等效的，它们生成相同的结果。处理多字节字符集时，如果 **-b** 和 **-n** 选项一起使用，它们的组合行为将非常类似，但与 **-c** 选项不同。

选项

选项解释如下：

list 整数 *byte* （ -b 选项）、 *character* （ -c 选项）或 *field* （ -f 选项）数字的逗号分隔列表（按升序排列），可选的 **-** 用来指示范围。例如：

1,4,7	位置 1、4 和 7。
1-3,8	位置 1 到 3 和位置 8。
-5,10	位置 1 到 5 和位置 10。
3-	位置 3 到最后一个位置。

-b *list* 根据字节列表提取。除非还指定了 **-n** 选项，否则每个所选字节均是输出。

-c *list* 根据 *list* 指定的字符位置提取（ -c 1-72 提取每行的前 72 个字符）。

-f *list* 其中 *list* 是假定为在文件中由定界符分隔的字段的列表（请参阅 **-d** ）；例如， **-f 1,7** 仅复制第一个和第七个字段。除非指定了 **-s** ，无字段定界符的行将按原样传递（对于表副标题非常有用）。

-d *char* **-d** 后的字符是字段定界符（仅限 -f 选项）。缺省值是 *tab* 。空格或其他对 Shell 具有特殊含义的字符必须进行引用。相邻字段定界符将定界空字段。 *char* 可以是国际代码集字符。

-n 不拆分字符。如果列表中范围的高端不是字符的最后一个字节，则该字符将不包含在输出中。但是，如果列表中范围的低端不是字符的第一个字节，则该字符 会包含在输出中。

-s 在使用 **-f** 选项时禁止无定界符的行。除非指定了 **-s** ，无定界符的行将按原样出现在输出中。

提示

使用 **grep** 可基于文本模式识别（使用正则表达式）从文件中提取文本。使用 **paste** 可以按分列格式逐行合并文件。要以不同的顺序重新排列表中的列，请使用 **cut** 和 **paste** 。有关详细信息，请参阅 *grep*(1) 和 *paste*(1) 。

外部语言环境影响

环境变量

LC_CTYPE 确定将文本解释为单字节和（或）多字节字符的方法。

如果未在环境中指定 **LC_CTYPE** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果 **LANG** 未指定或设置为空字符串，则将使用缺省的“C”而不是 **LANG** （请参阅 *lang*(5) ）。如果任一国际化变量中包含无效设置， **cut** 会假定所有国际化变量都缺省为“C”。请参阅 *environ*(5) 。

国际代码集支持

cut 支持单字节和多字节字符代码集。国际代码集字符可以在向 **-d** 选项提供的 *char* 中指定。 **cut** 按照在 **LC_CTYPE** 环境变量中指定的语言环境来识别国际代码集字符。

诊断信息

line too long
: 行长度不得超过 **LINE_MAX** 个字符或字段，包括换行符（请参阅 *limits*(5) ）。

bad list for b/c/f option
: 缺少 **-b** 、 **-c** 或 **-f** 选项，或者错误指定了 *list* 。如果行包含的字段少于 *list* 所需的字段，则不会出错。

no fields *list* 为空。

举例

用户 ID 到用户名的口令文件映射：

```
cut -d : -f 1,5 /etc/passwd
```

将环境变量 **name** 设置为当前登录名：

```
name=‘who am i | cut -f 1 -d " "‘
```

将包含任意长度的行的文件 **source** 转换为两个文件： **file1** 包含前 500 个字节（除非第 500 个字节处于多字节字符内）， **file2** 包含每一行的剩余部分：

```
cut -b 1-500 -n source > file1
cut -b 500- -n source > file2
```

警告

cut 不扩展制表符。如果要求进行制表符扩展，则通过 *expand*(1) 以管道传送文本。

退格字符的处理方式与其他任何字符均相同。要在 **cut** 处理退格字符前将其去除，请使用 **fold** 或 **col** 命令（请参阅 *fold*(1) 和 *col*(1) ）。

作者

cut 由 OSF 和 HP 联合开发。

另请参阅

grep(1)、 paste(1)。

符合的标准

cut：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

date - 显示或设置日期和时间

概要

date [**-u**]

date [**-u**] **+***format*

date [**-u**] [*mmddhhmm*[[*cc*]*yy*]]

date [**-a** [**-**]*sss*[*.fff*]]

说明

date 命令显示或设置当前 HP-UX 系统时钟的日期和时间。由于 HP-UX 系统按国际标准时间 (UTC) 运行， **date** 会根据您的 **TZ** 环境变量从（或向）本地标准或夏时制时间进行自动转换。请参阅下文“外部语言环境影响”中的环境变量。

选项

date 识别以下选项:

-u 采用国际标准时间 (UTC) 的输入值和输出值，在功能上等效于格林威治标准时间 (GMT) 而非本地时间。

-a [**-**]*sss*[*.fff*]
将时间缓慢地调整 *sss.fff* 秒（ fff 表示秒的分数部分）。该调整可以是正或负。系统时钟将走快或走慢，直至改动指定的秒钟数。

格式

date 命令有两种显示日期和时间的格式和一种设置日期和时间的格式。

date [**-u**]

显示当前日期和时间。对于除 **C** 缺省语言之外的所有语言，其输出与 **%c** 格式设置指令相同。请参阅下面的格式设置指令和“示例”。

date [**-u**] **+***format*

按照在 *format* 中指定的格式设置指令（零个或更多个格式设置指令和常规字符组成的字符串）显示日期和时间。如果它包含空白字符，请将其放在撇号或引号内。

请参阅下面的格式设置指令。

将所有常规字符按原样复制到输出字符串。

输出字符串始终用换行符终止。

如果指定 **+** 并省略 *format* ，则仅输出换行符。

date [**-u**] [*mmddhhmm*[[*cc*]*yy*]]

将 HP-UX 系统时钟设置为指定的日期和时间。您必须拥有超级用户特权。

如果包括 **-u** 选项，则假定指定的日期和时间是国际标准时间 (UTC)。

数值参数按照两位数对从左向右进行解释，如下所示：

mm 表示月份的数字 [**01-12**] 。
dd 表示一月中的某一日的数字 [**01-31**] 。
hh 表示小时的数字（24 小时制） [**00-23**] 。
mm 表示分钟的数字 [**00-59**] 。
cc 世纪减 1 [**19-20**] 。
yy 年份的后两位数字 [**70-99**, **00-37** （1970-1999、2000-2037）] 。如果省略，则使用当前年份。

如果尝试向后设置日期， **date** 将生成警告。

do you really want to run time backwards?[yes/no]

键入 **yes** （或您的语言环境的等效表示法）可向后设置时钟；其他任何输入都将取消命令。

当 **date** 用于设置日期时，一对数据更改记录将写入文件 **/var/adm/wtmps** 。

（仅适用于 XPG4） 。在向后设置日期时不生成警告。

格式设置指令

以下格式设置指令（显示时不带可选的字段宽度和精度规范）将替换为指定的字符。如果指令不是下列指令之一，则结果为未定义。

数字、字符和单词的输出取决于语言（或语言环境）设置。请参阅下文“外部语言环境影响”中的环境变量。

这些示例假定 **date** 命令的执行时间是 1994 年 1 月 12 日星期三下午 7:45:58（太平洋标准时间） ，并使用 **C** 缺省语言。

%a 缩写的星期名称。例如 **Wed** 。

%A 完整的星期名称。例如 **Wednesday** 。

%b 缩写的月份名称。例如 **Jan** 。

%B 完整的月份名称。例如 **January** 。

%c 当前的日期和时间形式。例如 **Wed Jan 12 19:45:58 1994** 。

%C 作为两位十进制数字的世纪（年份除以 100 并截断到整数） [**00-99**] 。例如 **19** 。

%d 作为两位十进制数字的一月中的某一日 [**01-31**] 。例如 **12** 。

%e 作为双字符十进制数字的一月中的某一日，带有前导空格填充 [**" 1"**- **"31"**] 。例如 **12** 。

%E 组合的帝号/纪元号和年份。

%H 作为两位十进制数字的小时（24 小时制） [**00-23**] 。例如 **19** 。

%I 作为两位十进制数字的小时（12 小时制） [**01-12**] 。例如 **07** 。

%j 作为三位十进制数字的一年中的天数 [**001-366**] 。例如 **012** 。

%m 作为两位十进制数字的月份 [**01-12**] 。例如 **01** 。

%M 作为两位十进制数字的分钟 [**00-59**] 。例如 **45** 。

%n 换行符。

%N 帝号/纪元号。

%o 帝号/纪元号年份。

%p 等效于 AM 或 PM。例如 **PM** 。

%R 时间格式 %H:%M

%S 作为两位十进制数字的秒钟（支持可能的闰秒） [**00-61**] 。例如 **58** 。

%t 制表符。

%u 作为一位十进制数字的星期 [**1-7**] 。例如 **3** 。

%U 作为两位十进制数字的一年的第几周（星期日作为一周的第一天） [**00-53**] 。一年中第一个星期日前的所有天均被视作第 **00** 周。例如 **02** 。

%V 作为两位十进制数字的一年的第几周（星期一作为一周的第一天） [**01-53**] 。如果包含 1 月 日的那一周在新的一年有四天或更多天（1 月 1 日是星期四或更早），则将其指定为第 **01** 周；否则（1 月 1 日是星期五或更晚），则将其指定为前一年的最后一周，它的下一周是第 **01** 周。例如 **02** 。

%w 作为一位十进制数字的星期 [**0-6**（星期日到星期六）] 。例如 **3** 。

%W 作为两位十进制数字的一年的第几周（星期一作为一周的第一天） [**00-53**] 。一年中第一个星期一前的所有天均被视作第 **00** 周。例如 **02** 。

%x 当前的日期形式。例如 **01/12/94** 。

%X 当前的时间形式。例如 **19:45:58** 。

%y 作为两位十进制数字的年份（不含世纪） [**00-99**] 。例如 **93** 。

%Y 作为四位十进制数字的年份（含世纪） [**1970-2037**] 。例如 **1994** 。

%Z 时区名称（在无法确定时区时无字符）。例如 **PST** 。

%% **%** 字符。

过时的指令

以下指令是为向后兼容提供的。建议使用前面所述的指令。

- **%D** 常用美制日期。例如 **01/12/94** 。改用 **%x** 或 **%m/%d/%y** 。
- **%F** 完整的月份名称。例如 **January** 。改用 **%B** 。
- **%h** 缩写的月份名称。例如 **Jan** 。改用 **%b** 。
- **%r** 12 小时美制时间。例如 **07:45:58 PM** 。改用 **"%I:%M:%S %p"** 。
- **%T** 24 小时美制时间。例如 **19:45:58** 。改用 **%X** 或 **%H:%M:%S** 。
- **%z** 时区名称（在无法确定时区时无字符）。例如 **PST** 。改用 **%Z** 。

修饰的格式设置指令

某些格式设置指令可以使用 **E** 和 **O** 修饰符进行修饰，为 **LC_TIME** 环境变量中指定的语言指定不同的格式或规范。

如果未指定或不支持相应的关键字 （**era** 、 **era_year** 、 **era_d_fmt** 和 **alt_digit** ），则使用未修饰的字段描述符值。以下命令：

LC_ALL=*language* **locale -ck era era_year era_d_fmt alt_digit**

以指定的 *language* 显示关键字和它们的值（请参阅 *locale*(1) ）。

- **%Ec** 适当的备用日期和时间形式。
- **%EC** 备用形式的基础年份的名称。
- **%Ex** 备用日期形式。
- **%Ey** 在备用形式中从 **%EC** 偏移（仅限年份）。
- **%EY** 完整的备用年份形式。
- **%Od** 采用备用数值符号的月中某日。
- **%Oe** 采用备用数值符号的月中某日，在可行时带前导空格字符。
- **%OH** 采用备用数值符号的小时（24 小时制）。
- **%OI** 采用备用数值符号的小时（12 小时制）。
- **%Om** 采用备用数值符号的月份。
- **%OM** 采用备用数值符号的分钟。
- **%OS** 采用备用数值符号的秒钟。
- **%OU** 采用备用数值符号的一年中的第几周（星期日是一周的第一天）。

%Ow 采用备用数值符号的数字形式的星期 (Sunday=**0**) 。

%OW 采用备用数值符号的一年中的第几周（星期一是一周的第一天）。

%Oy 备用形式的年份（从 **%C** 偏移）。

字段宽度和精度

格式设置指令的初始 **%** 之后可以立即添加可选的字段宽度和精度规范，其顺序如下：

[**-**|**0**]*width* 十进制数字字符串 *width* 指定对转换结果进行右或左对齐的 最小字段宽度。缺省值是通过在左侧填充空格进行右对齐。如果字符串以“**-**”开头，则结果将通过在右侧填充空格进行左对齐。如果字符串以“**0**”开头，则结果将通过在左侧填充空格进行右对齐。

.*prec* 十进制数字字符串 *prec* 指定 **d** 、 **H** 、 **I** 、 **j** 、 **m** 、 **M** 、 **o** 、 **S** 、 **U** 、 **w** 、 **W** 、 **y** 和 **Y** 数值指令可显示的 最小数字位数。如果指令提供的位数少于精度指定的位数，则将使用前导零进行扩展。

prec 指定 **a** 、 **A** 、 **b** 、 **B** 、 **c** 、 **D** 、 **E** 、 **F** 、 **h** 、 **n** 、 **N** 、 **p** 、 **r** 、 **t** 、 **T** 、 **x** 、 **X** 、 **z** 、 **Z** 和 **%** 文本指令中可使用的 最大字符数。如果指令提供的字符数大于精度指定的字符数，则将在右侧截断多余的字符。

如果没有为 **d** 、 **H** 、 **I** 、 **m** 、 **M** 、 **S** 、 **U** 、 **W** 或 **y** 指令指定字段宽度或精度，则缺省值是 **.2** ；对于 **j** 指令，缺省值是 **.3** ；对于 **Y** ，缺省值是 **.4** ；对于 **w** ，缺省值是 **.1** 。

外部语言环境影响

环境变量

LC_CTYPE 确定如何将 *format* 字符串内的字节解释为单字节和（或）多字节字符。

LC_NUMERIC 确定用于为在输出中生成数字的指令形成数字的字符。所使用的字符是由 **alt_digit** 定义的字符（请参阅 *langinfo*(5)中的 *locale*(1) 和 **ALT_DIGIT** ）。

LC_TIME 确定 **date** 命令输出的日期和时间字符串的内容（例如 **%a** 指令生成的星期名称）和格式（例如 **%X** 指令生成的当前时间形式）。

LC_MESSAGES 确定显示消息（非日期和时间字符串）的语言。

如果未指定 **LC_CTYPE** 、 **LC_NUMERIC** 、 **LC_TIME** 或 **LC_MESSAGES** ，或者其值为空，则缺省为 **LANG** 的值。

如果未指定 **LANG** ，或者其值为空，则缺省为 **C** （请参阅 *lang*(5) ）。

如果任一国际化变量包含无效设置，则所有国际化变量将缺省为 **C** （请参阅 *environ*(5) ）。

TZ 确定 UTC 系统时间和用户本地时区时间之间的转换。请参阅 *environ*(5) 和 *tztab*(4) 。 **TZ** 还确定 **date** 命令输出的日期和时间字符串的内容（即 **%z** 和 **%Z** 指令生成的时区名称）。

如果未设置 **TZ** ，或者其值设置为空字符串，其缺省值是 **EST5EDT** 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

可能会显示以下消息。

bad conversion

指定的日期/时间在句法上不正确。对照用法进行检查，并查看每个数字对的正确范围。

bad format character - *c*

字符 *c* 不是有效的格式指令、字段宽度指定符或精度指定符。

do you really want to run time backwards?[yes/no]

您指定的日期和（或）时间早于当前时钟值。键入 **yes** （或您的语言环境的等效表示法）可向后设置时钟；其他任何输入都将取消命令。

no permission

需要拥有超级用户特权才能更改日期。

举例

不同语言的日期

显示日期。在该示例中， **TZ** 环境变量包含 **PST8PDT** ，语言环境变量按照说明进行设置。

date → **Fri Aug 20 15:03:37 PDT 1993** ← **C** (default)
date -u → **Fri Aug 20 22:03:37 UTC 1993** ← **C** (default)
date → **Fri, Aug 20, 1993 03:03:37 PM** ← **en_US.roman8** (U.S. English)
date → **Fri. 20 Aug, 1993 03:03:37 PM** ← **en_GB.roman8** (U.K. English)
date → **20/08/1993 15.47.47** ← **pt_PT.roman8** (Portuguese)

设置日期

将日期设置为 10 月 8 日上午 12:45。

date 10080045

显示设置格式的日期

使用某种格式显示当前日期和时间。请注意，在这种格式中由于存在空白字符而使用了引号。

date "+DATE: %m/%d/%y%nTIME: %H:%M:%S"

输出类似于以下内容：

DATE: 10/08/87
TIME: 12:45:05

使用本地语言转换显示已设置格式的日期

日期如以上“设置日期”示例进行设置， **LC_TIME** 设置为 **de_De.roman8** （德语）：

date +'%-4.4h %2.1d %H:%M'

生成如下输出：

Okt 8 12:45

其中的月份字段为四个字符长，左对齐，当月份名称短于四个字符时在右侧填充空格。日期字段为两个字符长，禁止前导零。

警告

原来的 HP-UX 格式指令 **A** 已更改为 **W** ，以兼容 ANSI。

如果在系统以多用户模式运行时对日期进行更改，可能会破坏用户计划的和对时间敏感的程序与进程。此外，更改日期可能会导致 *make*(1) 以及 SCCS 和 *cron*(1M) 子系统出现意外的行为。向后设置日期之前，应先强行终止 **cron** 守护程序，然后将其重新启动。如果在时钟设置错误时创建了 delta 版，应该使用 **val** 命令检查 SCCS 文件（请参阅 *val*(1) ）。

将来的版本中可能会删除以下格式设置指令： **%E** 、 **%F** 、 **%o** 、 **%z** 。

目前支持的最大日期是国际标准时间 UTC 2037 年 12 月 31 日 23:59:00。

作者

date 由 AT&T 和 HP 联合开发。

文件

/var/adm/wtmps

另请参阅

locale(1)、 stime(2)、 ctime(3C)、 strftime(3C)、 tztab(4)、 environ(5)、 lang(5)、 langinfo(5)。

符合的标准

date：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

dc - 桌面计算器

概要

dc [*file*]

说明

dc 是一个任意精度的运算程序包。它通常对十进制整数执行运算，但是也可以指定输入基数、输出基数和要保留的小数位数。（请参阅 *bc*(1)，它是 **dc** 的预处理程序，可以提供中辍符号和类似 C 语言的语法以实现功能。 **bc** 还可为程序提供合理的控制结构）。 **dc** 的整体结构是堆栈式（逆波兰式）计算器。如果指定了参数，则会先从该文件中获取输入，直到文件结尾，然后再从标准输入中获取。标准输入的文件结束标志或 **q** 命令可以终止 dc。该命令可以识别下列结构：

number 该数的值被推入堆栈中。数是由数字 **0-9** 或 **A-F** 组成的完整字符串。可在字符串之前加一个下划线 (_) 以输入负数。数可以包含小数点。

+ - / * % ^ 堆栈中的前两个值进行加 (**+**)、减 (**-**)、乘 (*****)、除 (**/**)、求余 (**%**) 或取幂 (**^**) 运算。这两个条目会从堆栈中弹出；计算的结果将被推入到它们所在的位置。指数的任何小数部分将被忽略，同时会发出警告。余数是根据当前的比例因子计算的；它不是整数模数函数。如果比例因子为 1，则 **7 % 3** 的结果为 .1（十分之一），因为 **7 / 3** 是 2.3，余数是 .1。

s*x* 弹出堆栈的顶部数据并将其存储到名为 *x* 的寄存器中，此处的 *x* 可以是任意字符。如果 **s** 为大写，则会将 *x* 视为堆栈且值被推入到堆栈中。

l*x* 将寄存器 *x* 中的值推入堆栈中。寄存器 *x* 不会改变。所有寄存器都从零值开始。如果 **l** 为大写，则会将 *x* 视为堆栈，且其顶部值将被弹出并推入到主堆栈中。

d 复制堆栈顶部的值。

p 输出堆栈顶部的值。顶部的值保持不变。 **P** 将堆栈顶部的值解释为 ASCII 字符串，然后删除并输出该字符串。

f 输出堆栈中的所有值。

q 退出程序。如果正在执行字符串，则递归级别将弹出为 2。如果 **q** 为大写，则堆栈顶部的值将弹出，且字符串执行级别也将作为该值弹出。

x 将堆栈的顶部元素视为字符串，并将其作为 **dc** 命令的字符串执行。

X 以比例因子替代堆栈顶部的数。

[...] 将方括号中的 ASCII 字符串放置在堆栈的顶部。通过使用成对嵌套的方括号可以嵌套字符串。

<*x* >*x* =*x*

!<*x* **!>*x*** **!=*x***
: 弹出并比较位于堆栈顶部的两个元素。如果这两个元素遵循规定的关系，则将计算寄存器 *x* 。

v
: 用堆栈中顶部元素的平方根替代该元素。此时参数的任何现有的小数部分将参与计算，但是比例因子会被忽略。

!
: 将命令行的其余部分解释为 HP-UX 系统命令（除非接下来的字符是 **<** 、 **>** 或 **=** ，此时使用了上述适当的关系运算符）。

c
: 弹出堆栈中的所有值。

i
: 弹出堆栈顶部的值，并将其用作进一步输入中的数字基数。

I
: 将输入基数推入到堆栈顶部。

o
: 弹出堆栈顶部的值，并将该值用作进一步输出中的数字基数。关于输出基数的注释，请参阅下文。

O
: 将输出基数推入到堆栈顶部。

k
: 弹出堆栈顶部的值，并将该值用作非负比例因子：将在输出中显示适当的小数位数，并在乘法、除法和求幂运算期间保留。如果同时更改了比例因子、输入基数和输出基数，则它们之间的交互作用是合理的。

K
: 将比例因子推入到堆栈的顶部。

z
: 将堆栈层推入到堆栈中。

Z
: 以堆栈顶部数的长度替代该数。

?
: 从输入源（通常是终端）中获取一行输入并执行该行。

; 和 **:**
: **bc** 用其进行数组操作。

Y
: 生成 **dc** 自身的调试输出。

输入基数可以是任何数，但是对于输入，只有数字 0-9 和 A-F 是可用的，这样就限制了在范围 1-16 之外的基数的效用。可在任何基数中使用所有可能的 16 个数字，这些数字始终取其常规值。

输出基数可以是任意数。范围 2-16 内的基数可生成“常规”结果，其中字母 A-F 代表 10 到 16 之间的值。基数 0 和 1 将生成由 **1** 组成的字符串，该字符串的长度是该数的值。基数 -1 生成由 **d** 组成的类似字符串。其他基数的每位“数字”代表该数字序数的（多位数）十进制数。每个“数字”被标记为负基数。“数字”由空格分隔。指定了输出基数的定义之后，命令 **Op** 始终会输出“10”（以适合基数的表示形式表示）； **O1-p** 可输出关于该输出基数的有用信息。

诊断信息

x **is unimplemented** 这里 *x* 是一个八进制数。

stack empty	堆栈中的元素不足而不能执行所要求的操作。
Out of space	可用列表已耗尽（数字太多）。
Out of headers	保留了过多的数。
Out of pushdown	堆栈中的项目过多。
Nesting Depth	嵌套执行的层过多。

举例

该示例将输出 n!（n 的阶乘）的前十个值：

```
[la1+dsa*pla10>y]sy
0sa1
lyx
```

另请参阅

bc(1)。

《Number Processing Users Guide》中的《DC:An Interactive Desk Calculator》教程。

名称

dd - 对（磁带）文件进行转换、重新分块、翻译和复制

概要

dd [*option*=*value*] ...

说明

dd 将指定的输入文件复制到指定的输出文件，同时进行可能的转换操作。缺省情况下，使用标准的输入和输出。可以指定输入和输出块大小，以便利用原始的物理 I/O。完成操作之后， **dd** 报告全部和部分输入和输出记录的数量。

选项

dd 可识别下列 *option*=*value* 对：

if=*file* 输入文件名；缺省值是标准输入。

of=*file* 输出文件名；缺省值是标准输出。输出文件是使用 **creat()** 所用的同一所有者和组创建的。

ibs=*n* 输入块大小为 *n* 字节；缺省值为 512。

obs=*n* 输出块大小为 *n* 字节；缺省值为 512。

bs=*n* 将输入和输出块大小设置为相同大小，以替换 **ibs** 和 **obs** 。如果未指定转换（ conv 选项），则该选项尤其有效，这是因为不需要进行核内复制。

cbs=*n* 转换缓冲区大小为 *n* 字节。

skip=*n* 在开始复制之前，跳过 *n* 个输入块。

iseek=*n* 在开始复制之前，跳过 *n* 个输入块。（这是 **skip** 选项的一个别名）。

seek=*n* 在复制之前，从输出文件开头跳过 *n* 个块。

oseek=*n* 在复制之前，从输出文件开头跳过 *n* 个块。（这是 **seek** 选项的一个别名）。

count=*n* 仅复制 *n* 个输入块。

files=*n* 复制并串连 *n* 个输入文件。仅当输入文件是一个磁带设备时，才应使用该选项。

conv=*value* [**,***value* ...]

其中 *value* 是下表中用逗号分隔的符号。

ascii 将 EBCDIC 转换为 ASCII。

ebcdic 将 ASCII 转换为 EBCDIC。

ibm 使用备用转换表将 ASCII 转换为 EBCDIC。

ascii 、 **ebcdic** 和 **ibm** 值是相互排斥的。

block 将每个以换行符终止的或以文件结束标志终止的输入记录，转换为具有固定长度（由 **cbs** 指定）的记录。将删除所有换行符，并使用空格字符将块填充到 **cbs** 大小。长度大于 **cbs** 的行将被截断；并报告截断的行（记录）的数量（请参阅下面的诊断信息）。

block 和 **unblock** 值是相互排斥的。

unblock 将固定长度的输入记录转换为可变长度的记录。对于每个输入记录，将读取 **cbs** 字节数，删除结尾空格字符，并追加一个换行符。

lcase 将大写输入字符映射为相应的小写字符。

lcase 和 **ucase** 值是相互排斥的。

ucase 将小写输入字符映射为相应的大写字符。

swab 交换每一对输入字节。

noerror 发生输入错误时，不要停止处理。如果还指定了 **sync** 转换符号，则缺失的输入将替换为空字节并正常处理；否则，就会从输出中忽略输入块。

notrunc 不要截断现有的输出文件。输出文件中未被 **dd** 调用所覆盖的块将被保留。

sync 将每个输入块填充到 **ibs** 大小。如果还指定了 **block** 或 **unblock** ，则使用空格字符进行填充；否则，使用空字节进行填充。

在需要使用大小的地方， *n* 表示一个以字节为单位的数值。可以使用下列形式指定数值：

n	表示 *n* 字节
*n***k**	表示 *n* 千字节 ($n \times 1024$)，
*n***b**	表示 *n* 个块 ($n \times 512$)，或
*n***w**	表示 *n* 个字 ($n \times 2$)。

要表示一个产品，请使用 **x** 分隔数值对。

如果指定了 **block** 、 **unblock** 、 **ascii** 或 **ebcdic** 转换，则使用 **cbs** 选项。如果指定了 **ascii** ，则将 *cbs* 字符放到转换缓冲区中，将其转换为 ASCII，删除结尾空格，并在将行发送到输出之前添加一个换行符。如果指定了 **ebcdic**，则将 ASCII 字符读取到转换缓冲区中，将其转换为 EBCDIC，并添加空格来补足大小为 *cbs* 的输出块。

外部语言环境影响

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

环境变量

下列环境变量会影响 **dd** 的执行：

LANG 在 **LC_ALL** 及相应变量（以 **LC_** 开头）未指定语言环境时确定语言环境。

LC_ALL 确定用于覆盖 **LANG** 或以 **LC_** 开头的任意环境变量设置的任何值的语言环境。

LC_CTYPE 变量确定将文本数据的字节序列解释为字符（单字节/多字节字符，大写/小写字符）所用的语言环境。

LC_MESSAGES 变量确定写入消息所用的语言。

返回值

退出值为:

0	成功完成。
>0	发生了错误。

诊断信息

完成操作之后， **dd** 报告输入和输出记录的数量:

$f+p$ **records in**	读取的全部和部分块的数量。
$f+p$ **records out**	写入的全部和部分块的数量。

如果指定了 **conv=block** ，并且至少有一个截断的块，则还报告截断记录的数量:

n **truncated records**

举例

将 EBCDIC 磁带（分为 10 个块，每个块上有 80 字节的 EBCDIC 卡片穿孔码）读取到一个名为 **x** 的 ASCII 文件:

```
dd if=/dev/rmt/c0t0d0BEST of=x ibs=800 cbs=80 conv=ascii,lcase
```

请注意原始磁带设备文件的使用。 **dd** 特别适用于原始物理设备上的 I/O，这是因为它允许以任意块大小进行读取和写入操作。

警告

某些设备（例如 1/2 英寸磁带）不能进行搜索操作。通过使用 *mt*(1) 或一些其他适当命令，可以在运行 **dd** 之前对此类设备进行定位。 **skip**、**seek**、**iseek** 和 **oseek** 选项都适用于这些设备。但是，在无法进行搜索操作的设备上，使用这些选项跳过块是很缓慢的，因为必须将块实际读取到磁带上的所需位置。

ASCII 和 EBCDIC 转换表是从 1968 年 11 月 颁布的 256 字符 ACM 标准中提取出来的。虽然 **ibm** 转换作为一个标准不被大家广泛接受，但是它对某些 IBM 输出序列约定则相当有效。目前还没有一个通用的解决方案。

只有在转换为 ASCII 时才插入换行符；只有在转换为 EBCDIC 时才进行填充。这些应该是单独的选项。

如果 **if** 或 **of** 指的是原始磁盘，则 **bs** 应该始终为磁盘扇区大小的倍数。缺省情况下， **bs** 为 512 字节。如果磁盘扇区大小不是 512 字节，则应该使用扇区大小的倍数来指定 **bs** 。字符专用（原始）设备文件应当始终用于设备。

因为 **dd** 无法预先确定转换后所需的空间大小，所以这完全由用户决定，以便确保目标文件、文件系统和（或）设备中有足够的空间来包含输出。

另请参阅

cp(1)、 mt(1)、 tr(1)、 disk(7)、 mt(7)。

符合的标准

dd：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

delta - 对 SCCS 文件进行 delta（更改）

概要

delta [**-r** *SID*] [**-s**] [**-n**] [**-g** *list*] [**-m** *mrlist*] [**-y** *comment*] [**-p**] *files*

说明

delta 命令用于将对 **get** 所检索的文件（称为 *g-file* 或生成的文件）所做的更改引入指定的 SCCS 文件。请参阅 *get*(1) 。

delta 对每个指定的 SCCS 文件进行 delta 更改。如果指定了目录，则 **delta** 认为指定了该目录中的每个文件，但是其中非 SCCS 文件（路径名中不以 **.s** 开头的最后部分）和不可读文件将被默默忽略。如果指定了 **-** 的名称，则读取标准输入（请参阅 “警告”）。认为标准输入的每行都是要处理的 SCCS 文件的名称。

delta 可能在标准输出上发出提示，具体取决于指定的某些选项和 SCCS 文件（请参阅下面的 **-m** 和 **-y** 选项）中可能存在的标志（请参阅 *admin*(1) ）。

选项

可选参数单独应用于每个指定的文件。

-r*SID* 唯一标识对 SCCS 文件进行的 delta 更改。仅当同一 SCCS 文件上用于编辑的两个或更多未完成的 **get** (**get -e**) 由同一用户（登录名）进行时，才需要使用此选项。使用 **-r** 选项指定的 *SID* 值可以是在 **get** 命令行上指定的 *SID* 或 **get** 命令报告要生成的 *SID* （请参阅 *get*(1) ）。如果指定的 *SID* 是不明确的或者需要它但在命令行上省略了它，则产生诊断结果。

-s 禁止在标准输出上发出所创建 delta 版的 *SID* 以及 SCCS 文件中插入、删除和未更改的行数。

-n 指定保留已编辑的 *g-file* （通常在完成 delta 版处理时删除它）。

-g*list* 指定在此 delta 版所创建的更改级别 (SID) 上访问文件时要 忽略的 delta 版的 *list* （有关 *list* 的定义，请参阅 *get*(1) ）。

-m[*mrlist*] 如果 SCCS 已设置 **v** 标志（请参阅 *admin*(1) ），则 必须提供修改请求 (MR) 编号作为创建新 delta 版的理由。

如果未使用 **-m** 且标准输入是终端，则在读取标准输入之前，在标准输出上发出提示 **MRs?** 。如果标准输入不是终端，则不发出提示。 **MRs?** 提示总是在 **comments?** 提示的前面（请参阅 **-y** 选项）。

列表中的 MR 由空格和（或）制表符分隔。 MR 列表以未转义的换行符结尾。

请注意，如果 **v** 标志具有一个值（请参阅 *admin*(1) ），则假定它是验证 MR 编号正确性的程序（或 Shell 过程）的名称。如果从 MR 编号验证程序返回的退出状态不为零，则 **delta** 假定 MR 编号根本无效并终止。

-y[*comment*] 用于说明进行 delta 更改的原因的任意文本。将空字符串视为有效的 *comment* 。

如果未指定 **-y** 且标准输入是终端，则在读取标准输入之前，在标准输出上发出提示 **comments?** 。如果标准输入不是终端，则不发出提示。注释文本以未转义的换行符结尾。

-p 导致 **delta** 在应用 delta 版之前和之后打印（在标准输出上，以 *diff*(1) 格式） SCCS 文件差异。

外部语言环境影响

环境变量

LC_CTYPE 用于确定将文本解释为单字节和（或）多字节字符的方法。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或 **LC_MESSAGES** ，或者将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量中包含无效设置，则 **delta** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

可使用 *sccshelp*(1) 获得进一步说明。

警告

SCCS 文件可以是任意长度的，但是文本文件自身中的行数不能超过 99 999 行。

不能将以 ASCII SOH 字符（八进制的 001）开头的行放在 SCCS 文件中，除非将 SOH 转义。此字符对于 SCCS 具有特殊的含义（请参阅 *sccsfile*(4) ），并将导致错误。

在 **get** 生成大量数据时，应该避免许多 SCCS 文件的 **get** （后跟那些文件的 **delta** ）。相反，应该使用多个 **get/delta** 序列。

如果在 **delta** 命令行上指定了标准输入 **(-)** ，则 **-m** （如有必要）和 **-y** 选项也 必须存在。如果省略这些选项，则会导致错误。

注释可以是多行注释。注释的最大长度（所有注释行的总长度）不能超过 1024 字节。注释中所有行的长度均不应超过 1000 字节。

文件

下面列出的所有辅助文件（ *g-file* 除外）都是与 *s-file* 在同一目录中创建的（请参阅 *get*(1) ）。 *g-file* 是在用户的工作目录中创建的。

g-file 在执行 **delta** 前已经存在；在完成 **delta** 后被删除（除非指定了 **-n** ）。

p-file	在执行 **delta** 前已经存在；在完成 **delta** 后可能存在。
q-file	在执行 **delta** 期间创建；在完成 **delta** 后被删除。
x-file	在执行 **delta** 期间创建；在完成 **delta** 后被重命名为 SCCS 文件。
z-file	在执行 **delta** 期间创建；在执行 **delta** 期间被删除。
d-file	在执行 **delta** 期间创建；在完成 **delta** 后被删除。
/usr/bin/bdiff	用于计算 **get** 所检索的文件与 *g-file* 之间差异的程序。

另请参阅

admin(1)、 bdiff(1)、 cdc(1)、 get(1)、 sccshelp(1)、 prs(1)、 rmdel(1)、 sccsfile(4)。

符合的标准

delta：SVID2、SVID3、XPG2、XPG3、XPG4

名称

deroff - 删除 nroff、tbl 和 neqn 结构

概要

deroff [**-m***x*] [**-w**] [**-i**] [*file* ...]

说明

deroff 按顺序读取每一个 *file* ，并删除所有 **nroff** 请求、宏调用、反斜杠结构、 **neqn** 结构（在 **.EQ** 行和 **.EN** 行之间以及定界符之间 — 请参阅 *neqn*(1) ）和 **tbl** 说明（请参阅 *tbl*(1) ），将其替换为空白（空白字符和空白行），然后将在标准输出上写入文件的剩余部分。 **deroff** 跟踪包括文件链（ .so 和 **.nx nroff/troff** 格式化命令）；如果已包括某个文件，则将忽略命名该文件的 **.so** ，而命名该文件的 **.nx** 将终止执行。如果未提供输入文件， **deroff** 将读取标准输入。

-m 选项之后可接 **m** 、 **s** 或 **l** 。通过 **-mm** 选项，在解释宏时可仅输出正在运行的文本（即，不输出宏行中的文本）。 **-ml** 选项强制 **-mm** 选项，并且还致使删除与 **mm** 宏关联的列表。

如果给定了 **-w** 选项，输出就是单词列表，每行列出一个“单词”，其他所有字符均被删除。否则，输出将遵循原来的内容，只进行上述的删除。在文本中，“单词”是任意多字节字符串或任意包含至少两个字母并由字母、数字、 & 号和撇号 (**'**) 组成的字符串。但在宏调用中，“单词”是多字节字符串或以至少两个字母开头并总共包含至少三个字母的字符串。定界符是字母、数字、撇号和 & 号之外的任意字符。尾随撇号和 & 号将从“单词”中删除。

如果指定了 **-i** 选项， **deroff** 将忽略 **.so** 和 **.nx nroff/troff** 命令。

外部语言环境影响

环境变量

LC_CTYPE 确定如何将文本和文件名解释为单字节和（或）多字节字符。请注意，使用 **-w** 选项时不识别多字节标点字符。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或 **LC_MESSAGES** 或将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 。

如果任一国际化变量包含无效设置，则 **deroff** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

警告

deroff 不是完整的 **nroff** 解释程序；因此它可能无法解释细微的结构。大多数这样的错误将导致输出过多（而不是过少）。

-ml 选项无法正确处理嵌套列表。

作者

deroff 由加州大学伯克利分校开发。

另请参阅

neqn(1)、 nroff(1)、 tbl(1)。

名称

dhcpv6client_ui - DHCPv6 客户端接口，用于从 DHCPv6 服务器请求配置参数。

概要

/usr/bin/dhcpv6client_ui -m *interface_name* [**-n** *no_of_ip_addresses*] [**-o** *config_options*]

/usr/bin/dhcpv6client_ui -o *config_options*

/usr/bin/dhcpv6client_ui -m *interface_name* [**-R**]

/usr/bin/dhcpv6client_ui -r *interface_name*

/usr/bin/dhcpv6client_ui -v

说明

dhcpv6client_ui 表示接口，用户可以通过该接口与客户端守护程序联系，以便从服务器获得 IP 地址和其他配置参数。缺省配置参数被指定为调用 DHCPv6 客户端守护程序时的命令行选项。

当 **dhcpv6client_ui** 请求 IP 地址或其他配置参数时，客户端守护程序从服务器获取这些参数，并将其存储在 **/etc/dhcpv6client.data** 文件中。

dhcpv6client_ui 在配置客户端守护程序的同一台计算机上执行。确保在执行 **dhcpv6client_ui** 之前已经启动并正在运行客户端守护程序。

选项

dhcpv6client_ui 支持下列选项：

-m *interface_name* 要求客户端守护程序从 DHCPv6 服务器守护程序获取指定接口 *interface_name* 的 IP 地址。

-n *no_of_ip_addresses* 指定一个接口的 IP 地址号。

该选项始终与 **-m** 选项一起使用。

该选项不能与 **-R** 选项一起使用。

-o *config_options* 指定客户端守护程序必须从服务器守护程序请求的其他配置参数。客户端可以请求的配置参数如下所示：

dns_sa 获取 DNS 服务器地址

dns_sx 获取 DNS 后缀

ntp_sa 获取 NTP 服务器地址

nis_dn 获取 NIS 域名

nis_sa 获取 NIS 服务器地址

nispcl_dn 获取 NIS+ 客户端域地址

nisp_sa	获取 NIS+ 服务器地址
slp_da	获取 SLP 目录代理 (DA) 地址及其范围
slp_ss	获取 SLP 服务范围
tz	获取时区信息
all	获取上面列出的所有参数
default	获取 dhcpv6 客户端守护程序支持的所有参数

-r *interface_name* 释放分配给指定接口 *interface_name* 的 IPv6 地址。

-v 输出客户端守护程序的版本信息。

-R 通知客户端守护程序使用已获取的接口 IP 地址，而不是从服务器请求新的 IP 地址。

该选项必须与 **-m** 选项一起使用。

该选项不能与 **-n** 选项一起使用。

返回值

dhcpv6client_ui 在成功时返回 0，在失败时返回 1。

举例

dhcpv6client_ui 为 **lan0** 接口获取两个 IP 地址：

dhcpv6client_ui -m lan0 -n 2

dhcpv6client_ui 获取 **lan0** 接口的两个 IP 地址及其他配置参数：

dhcpv6client_ui -m lan0 -n 2 -o dns_sa dns_sx

文件

/etc/dhcpv6client.data 从服务器守护程序获取的所有数据都会保存到该文件中。

作者

dhcpv6client_ui 由 HP 开发。

另请参阅

dhcpv6clientd(1M)、 dhcpv6d(1M)。

名称

diff - 文件和目录差异比较程序

概要

diff [**-C** *n*] [**-S** *name*] [**-lrs**] [**-bcefhintw**] *dir1 dir2*

diff [**-C** *n*] [**-S** *name*] [**-bcefhintw**] *file1 file2*

diff [**-D** *string*] [**-biw**] *file1 file2*

说明

比较目录

如果两个参数都是目录，则 **diff** 按名称对目录的内容进行排序，然后对每个目录中同名但有差异的文本文件运行常规的文件 **diff** 算法（如下所述）。将列出有差异的二进制文件、公共子目录和仅出现在一个目录中的文件。在比较目录时，可识别下列选项：

-l 长输出格式；用管道通过 **pr** 传输每个文本文件 **diff** 的结果，以对它进行分页（请参阅 *pr*(1) ）。在报告所有的文本文件差异后，将记住和汇总其他差异。

-r 以递归方式将 **diff** 应用于遇到的公共子目录。

-s **diff** 报告完全相同的文件，如果文件不同则不报告。

-S *name* 在已排序目录的中间开始目录的 **diff** 比较，从文件 *name* 开始。

比较文件

对常规文件运行该命令时，以及比较在目录比较期间有差异的文本文件时， **diff** 指出必须更改文件中的哪些行才能使它们一致。 **diff** 通常会找到最小的充足文件差异集。但是，它可能会被包含字符非常少的行或其他情况误导。如果 *file1* 和 *file2* 都不是目录，则可以将任何一个指定为 **-** ，在这种情况下将使用标准输入。如果 *file1* 是一个目录，则使用该目录中与 *file2* 同名的文件（反之亦然）。

有几个输出格式选项。缺省的输出格式包含的行类似于：

n1 **a** *n3*,*n4*
n1,*n2* **d** *n3*
n1,*n2* **c** *n3*,*n4*

这些行与 **ed** 命令类似，可将 *file1* 转换为 *file2* 。字母后的数字与 *file2* 有关。事实上，通过将 **a** 与 **d** 交换并向后阅读，同样可以确定如何将 *file2* 转换为 *file1* 。与在 **ed** 中一样，会将完全相同的对（其中 *n1* = *n2* 或 *n3* = *n4* ）简写为单个数字。

上述行中每一行的后面显示第一个文件中标有 **<** 的所有受影响行，接着是第二个文件中标有 **>** 的所有受影响行。

除了 **-b** 、 **-w** 、 **-i** 或 **-t** 可以与任何其他选项同时指定外，以下选项是相互排斥的：

-e 为适于通过 *file1* 重新创建 *file2* 的 **ed** 编辑器生成一个包含 **a** 、 **c** 和 **d** 命令的脚本。在使用 **-e** 比较目录时会将额外命令添加到输出中，以便使生成的 Shell 脚本可将两个目录共有的文本文件从其在 *dir1* 中的状态转换为在 *dir2* 中的状态。

-f 生成与使用 **-e** 选项时类似的脚本，该脚本不能用于 **ed** ，但更易于阅读。

-n 生成与使用 **-e** 选项时类似的脚本，但顺序相反，且带有每个插入或删除命令所更改的行的数目。这是 **rcsdiff** 使用的形式（请参阅 *rcsdiff*(1)）。

-c 生成一个含有 3 行上下文的差异列表。 **-c** 对输出格式稍做修改：输出格式以所涉及文件的标识开头，后跟其创建日期，接着是由大约包含十二个星号 (*) 的行分隔的各处更改。从 *file1* 中删除的行标有 **-** ，添加到 *file2* 的行标有 **+** 。从一个文件更改到另一文件中的行在两个文件中都标有 **!** 。在输出时，会将文件中位于 3 行内的更改组成一组。

-C *n* 与使用 **-c** 时具有相似的输出格式，但含有 *n* 行上下文。

-h 执行快速的、小范围的作业。此选项只在更改的范围较小且分隔明确时才起作用，但可用于具有无限长度的文件。

-D *string* 在标准输出中创建 *file1* 和 *file2* 的合并版本，且引入 C 预处理器控制，以便在未定义 *string* 的情况下使编译结果等效于编译 *file1* ，而在定义 *string* 的情况下使编译结果等效于编译 *file2* 。

-b 忽略末尾的空白（空格和制表符），并将其他空白字符串视为相同。

-w 忽略所有空白（空白字符和制表符）。例如， **if (a == b)** 和 **if(a==b)** 被视为相同。

-i 忽略大小写的区别。因此， **A** 与 **a** 相同。

-t 在输出行中扩展制表符。正常输出或 **-c** 输出会将一个或多个字符添加到每一行的前面。这样会导致原始源文件行的缩进不对齐情况，从而使输出列表难以解释。此选项可保留原始的源文件缩进形式。

外部语言环境影响

环境变量

LANG 可确定当 **LC_ALL** 和相应的环境变量（以 **LC_** 开头）都未指定语言环境时，语言环境类别将使用的语言环境。如果不设置 **LANG** 或将其设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5)）。

LC_CTYPE 可确定 **diff** 命令的空格字符，以及将文件内的文本解释为单字节和（或）多字节字符。

LC_MESSAGES 用于确定显示消息的语言。

如果任一国际化变量包含无效设置，则 **diff** 和 **diffh** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5)。

国际代码集支持

支持单字节字符代码集和多字节字符代码集，例外情况是 **diff** 和 **diffh** 不能识别多字节替代空格字符。

返回值

diff 在完成时返回以下退出值之一:

0 未找到任何差异。

1 找到了差异。

>1 发生了错误。

举例

以下命令将创建一个脚本文件 **script** ：

diff -e x1 x2 >script

将 **w** 添加到脚本的末尾以便保存文件：

echo w >> script

然后，可以使用该脚本文件，通过 **ed** 编辑器从文件 **x1** 创建文件 **x2** ，如下所示：

ed x1 < script

以下命令将生成差异输出，在有差异的行前后各显示 2 行上下文信息：

diff -C2 x1 x2

以下命令将忽略所有空白和制表符，且忽略大小写的区别。

diff -wi x1 x2

警告

编辑 **-e** 或 **-f** 选项生成的脚本，与创建由单个点 (**.**) 组成的行相近，是不现实的。

在指定了 **-b** 、 **-w** 或 **-i** 选项的情况下比较目录时， **diff** 首先以与 **cmp** 相同的方式比较文件，然后在目录不同时运行 **diff** 算法。如果文件除了不重要的空白字符串或大小写差异之外完全相同，这可能导致错误地输出少量的差异。

缺省算法要求分配的内存大约是文件大小的六倍。如果没有足够的内存可用于处理较大文件，则可以使用 **-h** 选项或 **bdiff** （请参阅 *bdiff*(1) ）。

对于其他选项，如果没有足够的内存，可以增大 **swap** 或 **maxdsiz** 值。

在使用 **-r** 选项对目录运行 **diff** 时，它将以递归方式向下遍历子树。在比较多级深层目录时，可能需要比系统上当前可用内存更多的内存。所需的内存量取决于递归深度和文件大小。

作者

diff 由 AT&T、加州大学伯克利分校和 HP 开发。

文件

/usr/lbin/diffh 由 **-h** 选项使用

另请参阅

bdiff(1)、 cmp(1)、 comm(1)、 diff3(1)、 diffmk(1)、 dircmp(1)、 ed(1)、 more(1)、 nroff(1)、 rcsdiff(1)、 sccsdiff(1)、 sdiff(1)、 terminfo(4)。

符合的标准

diff：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

diff3 - 文件 3 个版本间的差异比较

概要

diff3 [-exEX3] *file1 file2 file3*

说明

diff3 比较一个文件的三个版本，并使用下列代码标记输出不一致的文本范围：

====	所有三个文件都不同
====1	*file1* 不同
====2	*file2* 不同
====3	*file3* 不同

将给定文件的指定范围转换为其他内容时，所需的更改类型由下列方式之一指定：

f **:** *n1* **a** 文本将被追加到文件 *f* 中的第 *n1* 行之后，其中 *f* = **1** 、 **2** 或 **3** 。

f **:** *n1* **,** *n2* **c** 从第 *n1* 行至第 *n2* 行范围内的文本将被更改。如果 *n1* = *n2* ，则该范围可以简写为 *n1*。

该范围的原始内容紧跟在 **c** 字符之后。当两个文件的内容完全相同时，将摒弃编号较小的文件的内容。

-e **diff3** 为 **ed** 编辑器生成脚本，可用于将 *file2* 和 *file3* 之间的所有更改都合并到 *file1* 中（请参阅 *ed*(1)）；即，合并通常由 **====** 和 **====3** 标记的更改。

-x 生成只合并标记为 **====** 的更改的脚本

-3 生成只合并标记为 **====3** 的更改的脚本

-E 生成脚本，该脚本将合并 *file2* 和 *file3* 之间的所有更改，但以不同方式处理重叠的更改（即，在常规列表中标记为 **====** 的更改）。两个文件中的重叠行将由 **<<<<<<** 和 **>>>>>>** 行括起来的编辑脚本插入。

-X 生成只合并标记为 **====** 的更改的脚本，但使用 **-E** 选项的方式处理这些更改。

下面的命令会将生成的脚本应用到 *file1* 。

```
(cat script; echo '1,$p') | ed - file1
```

外部语言环境影响

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

警告

由单个句点 (**.**) 组成的文本行将使 **-e** 无效。

对长度超过 64K 字节的文件无效。

文件

/var/tmp/d3*

/usr/lbin/diff3prog

另请参阅

diff(1)。

名称

diffmk - 对一个文件的两个不同版本之间的更改进行标记

概要

diffmk *prevfile currfile markfile*

说明

diffmk 对文件的前一版本和当前版本进行比较，并创建包括 **nroff/troff** “更改标记”命令的文件。 *prevfile* 为文件的先前版本的名称， *currfile* 为文件的当前版本的名称。 **diffmk** 生成 *markfile* ，其中包含 *currfile* 的所有行，以及插入的格式化程序“更改标记”(.mc) 请求。对 *markfile* 进行格式化时，由每行右边缘的 **|** 字符显示更改的或插入的文本。删除的文本的位置由单个 * 显示。

如果 **|** 字符和 * 字符均不适合，则可对 **diffmk** 的副本进行编辑，以更改这两个字符，因为 **diffmk** 是 Shell 脚本。

外部语言环境影响

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

比较 **nroff/troff** 文件的两个版本，并生成对更改进行了标记的文件的典型命令行如下:

diffmk prevfile currfile markfile; nroff markfile | pr

diffmk 还可以用于生成对更改进行了标记的 C 程序（或其他程序）列表。这种用途的典型命令行如下:

diffmk prevfile.c currfile.c markfile.c; nroff macs markfile.c | pr

其中 **macs** 文件包含:

```
.pl 1
.ll 77
.nf
.eo
```

.ll 请求可以指定不同的行长度，这取决于所输出的程序的特点。 **.eo** 请求可能仅 C 程序需要。

警告

出于审美方面的考虑，可以指示对一些输出进行手动调整。

diffmk 不对文本中的更改和格式化程序请求编码中的更改进行区分。因而，仅涉及格式化更改（如将源文本文件中的 **.sp** 替换为 **.sp 2** ），而实际文本无任何更改的文件差别，也能够生成更改标记。

尽管不太可能，但是某些格式化请求组合可以导致更改标记消失或标记过多。手动干预可能是需要的，因为各种格式化宏程序包和预处理器的细微程度可能超过 **diffmk** 的范围。 **tbl** 无法接受其输入中的 **.mc** 命令（请参阅 *tbl*(1) ），因此任何 **.mc** 出现在 **.TS** 范围内的请求都将以静默方式删除。如果此操作不合适，则可以对脚本进行更改，或在进行任何比较之前，对已在 **tbl** 预处理器中运行的两个文件运行 **diffmk** 。

diffmk 使用 **diff** ，因而它在文件大小和性能方面具有 **diff** 可能强制规定的相同限制（请参阅 *diff*(1) ）。尤其是，性能与文件大小呈非线性关系，处理大型文件（超过 1000 行）可能花费极长的时间。将文件拆分为几个较小的文件，可能是明智的做法。

diffmk 还使用 *ed*(1) 编辑器。如果对于 **ed** 而言，文件太大，则 **ed** 错误消息可能嵌入在文件中。同样，将文件拆分为几个较小的文件，可能是明智的做法。

另请参阅

diff(1)、 nroff(1)。

名称

dircmp - 目录比较

概要

dircmp [**-d**] [**-s**] [**-w***n*] *dir1 dir2*

说明

dircmp 检查 *dir1* 和 *dir2* ，并生成有关目录内容的各种表格信息。对于所有的选项都会生成对每个目录唯一的排序的文件列表。如果未输入选项，则输出一个已排序列表，以表明两个目录共有的文件名是否包含相同的内容。

-d 比较两个目录中同名文件的内容，并输出一个列表，说明为使同名文件内容一致而必须进行的修改。 *diff*(1) 中描述了列表的格式。

-s 消除有关相同文件的消息。

-w*n* 将输出行宽改为 *n* 个字符。缺省行宽为 72。

外部语言环境影响

环境变量

LC_COLLATE 用于确定输出的排序顺序。

如果未在环境中指定 **LC_COLLATE** 或将其设置为空字符串，则使用 **LANG** 作为缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省的“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **dircmp** 就会认为所有国际化变量都设置为“C”（请参阅 *environ*(5) ）。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

比较两个目录 **slate** 和 **sleet** ，并生成要更改的内容的列表以便使两个目录相同：

dircmp -d slate sleet

警告

该命令可能会从 X/Open 标准中删除。使用此命令的应用程序可能无法移植到其他供应商的系统上。推荐使用 **diff -R** 替代。

另请参阅

cmp(1)、diff(1)。

符合的标准

dircmp：SVID2、SVID3、XPG2、XPG3

名称

dmpxlt - 将 iconv 转换表转储成可读格式

概要

/usr/bin/dmpxlt [**-f** *output_filename*] [*input_filename*]

说明

dmpxlt 将编译版本的 **iconv** 代码集转换表转储为可读的 ASCII 格式，可以对该格式进行修改并将其用于 *genxlt*(1) 的输入，以便重新生成 *iconv*(1) 转换表。

选项

dmpxlt 识别以下选项：

-f *output_filename* 如果未选择此选项，则数据会发送至标准输出。

dmpxlt 将以指定格式创建输出文件，并提供两个代码集之间的文件代码映射， *genxlt*(1) 可以编辑并重新使用该映射，以便创建新的 *iconv*(1) 表。条目以十六进制表示。

外部语言环境影响

环境变量

LANG 为没有设置或设置为 null（空）的国际化变量提供了缺省值。如果 **LANG** 未设置或设置为 null（空），则会使用缺省值“C”（请参阅 *lang*(5)）。如果任一国际化变量中包含无效设置，则 **dmpxlt** 的行为类似于所有国际化变量都设为“C”。请参阅 *environ*(5)。

LC_ALL 如果设为非空字符串值，则会覆盖所有其他国际化变量的值。

LC_MESSAGES 确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息以及写入到标准输出的信息消息的格式和内容。

NLSPATH 确定消息目录的位置，以便处理 **LC_MESSAGES** 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

退出值如下：

0 成功完成。

>0 发生了错误。

举例

本示例将从表 **roma8=iso81** 创建源文件 *genxlt_input* ：

dmpxlt -f genxlt_input /usr/lib/nls/iconv/tables/roma8=iso81

文件

/usr/lib/nls/iconv/tables 所有表都必须安装在该目录下。

另请参阅

iconv(1)、 genxlt(1)、 iconv(3C)、 environ(5)、 lang(5)。

名称

dnssec-keygen - DNSSEC 密钥生成工具

概要

dnssec-keygen [**-a** *algorithm*] [**-b** *keysize*] [**-e**] [**-g** *generator*] [**-h**] [**-n** *nametype*]
[**-p** *protocol-value*] [**-r** *randomdev*] [**-s** *strength-value*] [**-t** *type*] [**-v** *level*] name

说明

dnssec-keygen 按照 RFC2535 中的定义为安全 DNS (DNSSEC) 生成密钥。它还会生成 RFC2845 中定义的事务签名 (TSIG) 中所使用的密钥。

参数

name 指定要为其生成密钥的域的名称。

选项

-a *algorithm* 这一选项用于指定加密算法。 *algorithm* 可以是 **RSAMD5** 、 **DH** 、 **DSA** ，也可以是 **HMAC-MD5** ，还可以使用 **RSA** ，它与 **RSAMD5** 等效。

用于判断加密算法的 *algorithm* 参数不区分大小写。 DNSSEC 将 DSA 指定为必需算法，而将 RSA 指定为推荐算法。 TSIG 的实现必须支持 HMAC-MD5。

-b *keysize* 这一选项用于确定密钥的位数。对密钥大小的选择具体取决于所用算法。

如果使用了 **RSA** 算法， *keysize* 必须在 512 位和 2048 位之间。

如果使用了 **DH** (Diffie-Hellman) 算法，则 *keysize* 必须在 128 位和 4096 位之间。

如果使用了 **DSA** (Digital Signature Algorithm) 算法， *keysize* 必须是 512 位和 1024 位之间的 64 的倍数。

如果使用了 **HMAC-MD5** 算法， *keysize* 必须在 1 位和 512 位之间。

-e 这一选项用于生成具有较大指数值的 RSA 密钥。

-g *generator* 这一选项用于创建 Diffie-Hellman 密钥。 **-g** 选项选择要使用的 Diffie-Hellman 生成器。支持的 *generator* 值只有 2 和 5。如果不支持任何 Diffie-Hellman 生成器，那么在可能的情况下，会使用 RFC2539 中的一个已知质数。否则将使用 2 作为生成器。

-h 这一选项可输出 **dnssec-keygen** 的选项及参数的摘要。

-n *nametype* 这一选项指定所生成的密钥的使用方式。

nametype 既可以是 **ZONE** 、 **HOST** 、 **ENTITY** ，也可以是 **USER** ，分别指明密钥将用于对区域、主机、实体或用户进行签名。在这样的环境中， **HOST** 和 **ENTITY** 是完全相同的。注意 *nametype* 不区分大小写。

-p *protocol-value*
: 这一选项将所生成的密钥的协议值设置为 *protocol-value* 。对于类型为 **USER** 的密钥，其缺省值为 2 (email)，其他所有类型密钥其缺省值则为 3 (DNSSEC)。该参数的其他可能值都在 RFC2535O 及其后续版本中列出。

-r *randomdev*
: 当系统没有 **/dev/random** 设备以生成随机数字时，该选项将覆盖 **dnssec-keygen** 使用随机数字生成密钥的过程。 **dnssec-keygen** 程序会提示您通过键盘输入，并且利用键击之间的时间间隔实现随机性。这一选项将 *randomdev* 作为随机数据的源。

-s *strength-value*
: 这一选项用于设置密钥的强度值。生成的密钥将使用 *strength-value* 对 DNS 资源记录进行签名。该值应该是一个 0 到 15 之间的数字。缺省的强度值为 0。密钥的强度字段当前未在 DNSSEC 中定义。

-t *type*
: 该选项指明该密钥是用于验证还是用于保密。 *type* 可以是 **AUTHCONF** 、 **NOAUTHCONF** 、 **NOAUTH** ，也可以是 **NOCONF** 。缺省值为 **AUTHCONF** 。如果类型为 **AUTHCONF**，则密钥可以同时用于验证和保密。将 *type* 设置为 **NOAUTHCONF** 即可指明密钥无法用于验证或保密。 **NOAUTH** 的值表示该密钥可用于保密，但不能用于验证。同样地， **NOCONF** 表示该密钥不能用于保密，但可以用于验证。

-v *level*
: 这一选项可用于使 **dnssec-keygen** 更为详细。随着调试/跟踪级别的提高， **dnssec-keygen** 所生成的操作报告也会越来越详细。缺省级别为 0。

生成的密钥

dnssec-keygen 结束时，会在标准输出中输出格式为 *Knnnn.+aaa+iiiii* 的字符串。这是所生成的密钥的一个识别字符串。这些字符串可作为 **dnssec-makekeyset** 实用程序的参数提供。

nnnn 部分是由 *name* 指定的以点结尾的域名。 DNSSEC 算法标识符是由 *aaa* 标识的： 001 表示 RSA，002 表示 Diffie-Hellman，003 表示 DSA，157 表示 HMAC-MD5。 *iiiii* 是一个用于标识密钥的 5 位数。

dnssec-keygen 创建两个文件。根据上面的密钥标识字符串调整文件名。文件名的形式是：

*Knnnn.+aaa+iiiii.***key** 和

*Knnnn.+aaa+iiiii.***private**。

其中分别包含密钥的公用与私用密钥部分。由 **dnssec-keygen** 生成的文件也遵循这一命名约定，这使得签名工具 **dnssec-signzone** 可以更容易地识别必须读取哪些文件，才能找到必需的密钥，以生成或校验证签名。

.key 文件包含一个 KEY 资源记录，它可插入到区域文件中，只需使用 **$INCLUDE** 语句即可。其中，密钥的私有部分位于 **.private** 文件中。该文件中包含所用加密算法的详细信息，以及所有相关参数：质数、幂数、模数及子质数等。出于一些显而易见的安全性考虑，并非所有人都对该文件有读取权限。密钥的私有部分是由 **dnssec-signzone** 使用的，用来生成签名，而公共部分则用来验证签名。 **.key** 和 **.private** 密钥文件都是根据对称加密算法生成的，例如 HMAC-MD5。尽管如此，公用密钥和私用密钥仍然是等效的。

举例

要生成域 **example.com** 的 768 位的 DSA 密钥，需要执行下列命令：

dnssec-keygen -a DSA -b 768 -n ZONE example.com

dnssec-keygen 已经输出了密钥标识字符串 **Kexample.com.+003+26160**，指明了一个标识符为 26160 的 DSA 密钥。它应该已创建了文件。

Kexample.com.+003+26160.key 并且

Kexample.com.+003+26160.private

分别包含所生成的 DSA 密钥的公用密钥和私用密钥。

文件

/dev/random

另请参阅

dnssec-makekeyset(1)、 dnssec-signkey(1)、 dnssec-signzone(1)、 RFC2535、 RFC2845、 RFC2539。

缺陷

公用密钥和私用密钥的命名约定有一点小小的缺陷。它对超过 236 个字符的域名不起作用，这是因为 .+*aaa*+*iiiii*.**private** 后缀会导致文件名过长，使大多数 UNIX 系统无法识别。

名称

dnssec-makekeyset - 用于生成 DNSSEC 密钥集

概要

dnssec-makekeyset [**-a**] [**-h** *help*] [**-s** *start-time*] [**-e** *end-time*] [**-t** *TTL*] [**-r** *randomdev*]
[**-p**] [**-v** *level*] *keyfile*...

说明

dnssec-makekeyset 从 **dnssec-keygen** 。创建的一个或多个密钥生成密钥集。它为某个区域创建包含 KEY 和 SIG 记录的文件（如果父区域可识别 DNSSEC，则该文件可以由该区域的父级进行签名）。

keyfile 应该是一个 **dnssec-keygen** 报告的密钥标识字符串；例如 **K***nnnn.+aaa+iiiii* ，其中 *nnnn* 是密钥的名称，*aaa* 是加密算法， *iiiii* 是密钥标识符。当存在要由 **dnssec-makekeyset** 组合为密钥集的几个密钥时，可以提供多个 *keyfile* 参数。

选项

-a 该选项用于验证所有的已生成签名。

-e *end-time* SIG 记录的到期日期可以由 **-e** 选项设置。请注意，在该上下文中，到期日期指定 SIG 记录何时不再有效，而不是指定何时将它们从名称服务器上的缓存中删除。

end-time 表示绝对日期或相对日期。 *YYYYMMDDHHMMSS* 格式的时间用于表示绝对日期和时间。

当 *end-time* 为 **+***N* 时，表示 SIG 记录将在其起始日期 *N* 秒之后到期。当 *end-time* 写为 **now+***N* 时，表示 SIG 记录将在其当前时间 *N* 秒之后到期。

如果没有为 SIG 记录设置到期日期，则 **dnssec-makekeyset** 缺省为从 SIG 记录的起始时间起 30 天后到期。

-h *help* 该选项用于显示随 **dnssec-makekeyset** 提供的选项的简短摘要。

-p 该选项用于指示 **dnssec-makekeyset** 在对密钥集进行自我签名时使用伪随机数据。这样比使用真正的随机数据进行签名速度更快，但是安全性更低。当信息源有限时，该选项可能很有用。

-r *randomdev* 可以使用 **-r** 选项指定随机数据的替代源。 *randomdev* 是用来获取随机数据的文件的名称。缺省情况下，如果该设备是可用的，则使用 **/dev/random** 。如果操作系统未提供该文件，且未使用 **-r** 选项，则 **dnssec-makekeyset** 将提示用户从键盘输入，并根据键击之间的时间获得一些随机数据。

-s *start-time* 对于密钥集中的任何 SIG 记录， SIG 记录变为有效的起始时间是使用 **-s** 选项指定的。 *start-time* 可以是绝对日期或相对日期。

绝对起始时间由 *YYYYMMDDHHMMSS* 格式的数表示；例如 **20000530144500** 表示 2000 年 5 月 30 日 14:45:00（国际标准时间）。

当 *start-time* 为 **+***N* 时，则提供了一个相对起始时间，即指定从当前时间开始的 *N* 秒。

如果没有提供 **-s** 选项，则会将当前日期和时间用作 SIG 记录的起始时间。

-t *TTL* **-t** 选项后跟生存时间参数 **TTL** ，它表示将分配给输出文件中已编译的 KEY 和 SIG 记录的 TTL 值。 **TTL** 是以秒表示的。如果未提供 **-t** 选项，则 **dnssec-makekeyset** 将输出警告并使用 3600 秒的缺省 TTL。

-v *level* 该选项可用于使 **dnssec-makekeyset** 输出更详细的信息。随着调试（或跟踪）级别 *level* 的提高， **dnssec-makekeyset** 将生成更加详细的关于正在执行的操作的报告。缺省级别为零。

如果 **dnssec-makekeyset** 成功，则它将创建格式为 *nnnn***.keyset** 的文件名。该文件包含域 *nnnn* 的 KEY 和 SIG 记录、 **dnssec-keygen** 创建该域的公用密钥和专用密钥时生成的密钥文件标识符中的域名部分。然后，可以将 **.keyset** 文件传输到父区域的 DNS 管理员，以便他们使用 **dnssec-signkey** 对内容进行签名。

举例

以下命令将为在 **dnssec-keygen** 联机帮助页中显示的 **example.com** 的 DSA 密钥生成密钥集。（请注意，反斜杠只是续行符，而不是 **dnssec-makekeyset** 命令语法的一部分）。

```
dnssec-makekeyset -t 86400 -s 20000701120000 -e +2592000 \
  Kexample.com.+003+26160
```

dnssec-makekeyset 将创建名为 **example.com.keyset** 、包含 **example.com** 的 SIG 和 KEY 记录的文件。这些记录将具有 86400 秒（1 天）的 TTL。 SIG 记录在 2000 年 7 月 1 日中午（国际标准时间）变为有效，并在 30 天（ 2592000 秒）后到期。

然后， **example.com** 的 DNS 管理员可能将 **example.com.keyset** 发送给 **.com** 的 DNS 管理员，以便他们可以对文件中的资源记录进行签名。这假定 **.com** 区域是识别 DNSSEC 的，而且这两个区域的管理员都具有相互验证以及安全地交换密钥和签名的某种机制。

文件

/dev/random

另请参阅

dnssec-keygen(1)、 dnssec-signkey(1)、 dnssec-signzone(1)、 RFC2535。

名称

dnssec-signkey - DNSSEC 密钥集签名工具

概要

dnssec-signkey [**-a**] [**-c** *class*] [**-e** *end-time*] [**-h**] [**-p**] [**-r** *randomdev*] [**-s** *start-time*]
[**-v** *level*] *keyset keyfile* ...

说明

dnssec-signkey 用于给子区域的密钥集签名。通常，它由 **dnssec-makekeyset** 实用程序生成的 **.keyset** 文件提供。通过它提供的机制，支持 DNSSEC 的区域可给支持 DNSSEC 的任何子区域的密钥签名。子区域的密钥集通过其父区域的区域密钥进行签名。

keyset 将是子区域的 **.keyset** 文件的路径名。

每个 *keyfile* 参数将是由父区域的 **dnssec-keygen** 报告的密钥标识字符串。这样，子区域的密钥可以由多个父区域密钥来签名。

选项

-a 该选项用于验证所生成的所有签名。

-c *class* 该选项指定密钥集的 DNS 类。当前仅支持 IN 类。

-e *end-time* 该选项指定生成的 SIG 记录过期的日期和时间。 *end-time* 表示绝对或相对的日期。*YYYYMMDDHHMMSS* 格式的时间用于表示绝对日期和时间。

当 *end-time* 为 **+***N* 时，表示 SIG 记录将在其起始日期 *N* 秒之后到期。如果 *end-time* 写为 **now+***N* ，表示 SIG 记录将在当前时间 *N* 秒之后到期。如果未指定 **end-time** ，则使用起始时间后的 30 天作为缺省值。

-h 该选项可使 **dnssec-signkey** 输出其命令行选项和参数的摘要。

-p 该选项可指示 **dnssec-signkey** 在进行密钥签名时使用伪随机数。

这样比使用真正的随机数据进行签名速度更快，但是安全性更低。当有多个要签名的子区域密钥集，或信息源有限时，该选项很有用。该选项也可用于无需进行深层破译保护的有效期较短的密钥及签名，例如在遇到安全危险前很早就被忽略的密钥。

-r *randomdev* 该选项覆盖 **dnssec-signkey** 的行为，在系统没有 **/dev/random** 设备来生成随机数时使用随机数作为密钥生成进程的种子。 **dnssec-signkey** 程序将提示用户从键盘输入，并根据键击之间的时间间隔来提供随机数。该选项使用 *randomdev* 作为随机数据源。

-s *start-time* 该选项指定生成的 SIG 记录开始生效的日期和时间。 *start-time* 可以是绝对日期或相对日期。

绝对起始时间由 *YYYYMMDDHHMMSS* 格式的数表示；例如 **20000530144500** 表示 2000 年 5 月 30 日 14:45:00（国际标准时间）。

当 *start-time* 给定为 **+***N* 时，则提供了一个相对起始时间，即指定从当前时间开始的 *N* 秒。如果未指定 **start-time** ，则使用当前时间。

-v *level* 该选项用于使 **dnssec-signkey** 输出更详细的信息。随着调试/跟踪级别的提高， **dnssec-signkey** 将生成更加详细的关于正在执行的操作的报告。缺省级别为零。

dnssec-signkey 成功完成时，将生成名为 *nnnn***.signedkey** 的文件，其中包含子区域 *nnnn* 的签名密钥。 **keyset** 文件中的密钥会已经由父区域的密钥或作为 **keyfile** 参数提供的密钥进行签名。该文件应发送到子区域的 DNS 管理员。当下次通过 **dnssec-signzone** 签名时，会安排将该文件的内容合并到区域文件中。父区域的 DNS 管理员应保留生成的 **signedkey** 文件的副本，这是因为在给父区域签名时将需要该文件。

举例

支持 DNSSEC 的 **.com** 区域的 DNS 管理员使用以下命令使得 **dnssec-signkey** 给 **example.com** （在 **dnssec-makekeyset** 联机帮助页中所示的示例中创建）的 **.keyset** 文件签名：

dnssec-signkey example.com.keyset Kcom.+003+51944

其中 **Kcom.+003+51944** 是在 **dnssec-keygen** 为 **.com** 区域生成密钥时所生成的密钥文件标识符。

dnssec-signkey 将生成名为 **example.com.signedkey** 的文件，该文件具有由 **com** 区域的区域密钥签名的 **example.com** 密钥。

文件

/dev/random

另请参阅

dnssec-keygen(1)、 dnssec-makekeyset(1)、 dnssec-signzone(1)、 RFC2535。

名称

dnssec-signzone - DNSSEC 区域签名工具

概要

dnssec-signzone [**-a**] [**-c** *cycle-time*] [**-d** *directory*] [**-e** *end-time*] [**-f** *output-file*] [**-h**]
[**-i** *interval*] [**-n** *ncpus*] [**-o** *origin*] [**-p**] [**-r** *randomdev*] [**-s** *start-time*] [**-t**]
[**-v** *level*] *zonefile keyfile* ....

说明

dnssec-signzone 用于对区域签名。要进行签名的区域的任何 **.signedkey** 文件连同区域签名所用的密钥都应位于当前目录中。

参数

zonefile 即未签名区域文件的名称。

keyfile 如果没有提供 *keyfile* 参数，则缺省的操作是使用该区域位于当前目录中的所有密钥。如果提供了特定的 *keyfile* 参数，则将限制 **dnssec-signzone** 仅使用这些密钥为区域签名。每个 *keyfile* 参数都是一个由 **dnssec-keygen** 创建的密钥的标识字符串。

如果要签名的区域具有任何安全子区域，则这些子区域的 **.signedkey** 文件必须位于 **dnssec-signzone** 使用的当前工作目录中。

选项

-a 该选项用于对 **dnssec-signzone** 生成的签名进行强制验证。缺省情况下不会验证签名文件。

-c *cycle-time*

当先前已签名的区域作为输入被传递给 **dnssec-signzone** 时，该选项可用于配置循环周期，以便对记录进行重新签名。循环周期是从当前时间开始的偏移量（秒）。如果 SIG 记录在循环周期后到期，则将保留该记录。否则，会认为该记录即将到期， **dnssec-signzone** 将删除该记录并生成一个新的 SIG 记录替代它。

-d *directory*

该选项用于在指定的目录中查找 **signedkey** 文件。

-e *end-time*

该选项用于设置 SIG 记录的到期时间。到期时间指定的是 SIG 记录何时不再有效，而不是何时从名称服务器的缓存中删除。 *end-time* 可以表示为绝对日期或相对日期。

YYYYMMDDHHMMSS 格式的时间用于表示绝对日期和时间。

当 *end-time* 为 **+***N* 时，表示 SIG 记录将在其起始时间 *N* 秒之后到期。

-f *output-file*

该选项用于覆盖缺省的已签名的区域文件 **zonefile.signed** （由 **dnssec-signzone** 签名）。

-h 该选项用于输出 **dnssec-signzone** 的选项及参数的简短摘要。

-i *interval* 当作为输入传递一个先前已签名的区域时，会对记录进行重新签名。 *interval* 选项将循环间隔指定为从当前时间开始的偏移量（秒）。如果 SIG 记录在循环间隔之后到期，则将保留该记录。否则，会认为该记录尚未到期并将替代它。

缺省的循环间隔为签名的结束时间与起始时间之间差值的四分之一。因此，如果既未指定 *end-time* ，也未指定 *start-time* ，则 **dnssec-signzone** 将生成有效期为 30 天的签名，其循环间隔为 7.5 天。这样，如果任何现有 SIG 记录在小于 7.5 天的时间间隔内到期，则这些记录将被替代。

-n *ncpus* 该选项可用于创建等于 *ncpus* 的工作线程，以便可以使用多个 CPU。如果不指定此选项，则 **named** 将尝试确定存在的 CPU 的数目，并为每个 CPU 创建一个线程。

-o *origin* 该选项可指定区域来源。如果不指定该选项，则将区域文件名视作 *origin* 。

-p 该选项可指示 **dnssec-signkey** 在进行密钥签名时使用伪随机数。这样比使用真正的随机数据进行签名速度更快，但是降低了安全性。当有多个要签名的子区域密钥集，或信息源有限时，该选项很有用。该选项也可用于无需同种程度破译保护的短期密钥及签名，例如在密钥早在其可能被泄露之前就被废弃的情况下。

-r *randomdev*

该选项会使 **dnssec-signzone** 使用随机数来促进区域签名的过程。如果系统不具有生成随机数的 **/dev/random** 设备，则 **dnssec-signzone** 程序会提示用户从键盘输入，并根据键击之间的时间间隔来提供随机数。该选项使用 *randomdev* 作为随机数据源。

-s *start-time*

该选项用于指定生成的 SIG 记录开始生效的日期和时间。 *start-time* 可以是绝对日期也可以是相对日期。

绝对起始时间由 *YYYYMMDDHHMMSS* 格式的数表示；例如 **20000530144500** 表示 2000 年 5 月 30 日 14:45:00（国际标准时间）。

当 *start-time* 为 **+N** 时，则提供了一个相对起始时间，即指定从当前时间开始的 *N* 秒。

如果没有提供 **-s** 选项，则会将当前日期和时间用作 SIG 记录的起始时间。

-t 该选项用于在命令完成时输出统计信息。

-v *level* 该选项用于使 **dnssec-signzone** 输出更详细的信息。随着调试/跟踪 *level* 的提高， **dnssec-signzone** 将生成更加详细的关于正在执行的操作的报告。缺省级别为零。

举例

下面的示例介绍了如何使用 **dnssec-signzone** 对 **example.com** 区域进行签名（通过在 **dnssec-keygen** 的联机帮助页的示例中生成的密钥）。该区域的区域文件是 **example.com** ，它与区域来源相同，所以无须使用 **-o** 选项设置来源。该区域文件包含 **dnssec-makekeyset** 创建的 **example.com** 的密钥集。通过使用 **$INCLUDE** 语句，将区域的密钥追加或合并到区域文件中。如果在父区域中存在 **.signedkey** 文件，即 **example.com.signedkey** ，则此文件应该存在于当前目录中。这就使父区域的签名可以被包括到 **example.com** 区域的签名版本中。

dnssec-signzone example.com Kexample.com.+003+26160

dnssec-signzone 将创建一个名为 **example.com.signed** 的文件，该文件是 *example.com* 区域的签名版本。然后可以在 **/etc/named.conf** 中的 **zone{}** 语句中引用该文件，使其可以被名称服务器加载。

文件

/dev/random

另请参阅

dnssec-keygen(1)、 dnssec-makekeyset(1)、 dnssec-signkey(1)、 RFC2535。

名称

domainname - 设置或显示网络信息服务域名

概要

domainname [*name_of_domain*]

说明

网络信息服务 (NIS) 使用域名来表示一组主机。不带参数时， *domainname* 显示的是 NIS 域的名称。只有超级用户才可以通过指定 *name_of_domain* 来设置域名。通常，可通过在配置文件 **/etc/rc.config.d/namesvrs** 中设置 **NIS_DOMAIN** 变量来设置域名。

相关内容

NIS 服务器使用 NIS 域名作为 **/var/yp** 的一个子目录名。为此， *name_of_domain* 不应是 **.** 或 **..** ，并且不应包含 **/** 。由于 NIS 域名可以长达 64 个字符，因此 *name_of_domain* 可能会超过给定文件系统中允许的最大文件名长度。如果超过了这个最大长度，那么子目录名就成为截断的 NIS 域名。

网络中所有 NIS 域的前 14 个字符必须是唯一的：应检查被截断的名称，以确认这些名称符合这一要求。

作者

domainname 由 Sun Microsystems, Inc. 开发。

另请参阅

ypinit(1M)、 getdomainname(2)、 setdomainna(2)。

名称

dos2ux、ux2dos - 转换 ASCII 文件格式

概要

dos2ux *file*...

ux2dos *file*...

说明

dos2ux 和 **ux2dos** 依次读取每个指定的 *file* ，将其写入标准输出，同时分别转换为 HP-UX 格式或 DOS 格式。每一命令的每个 *file* 可以是 DOS 格式或 HP-UX 格式。

如果名称中存在一个嵌入的冒号分隔符 (**:**) ，则表示该名称为 DOS 文件名；请参阅 *dosif*(4) 了解 DOS 文件的命名约定。

如果没有提供输入文件或者指定了参数 **-** ， **dos2ux** 和 **ux2dos** 将从标准输入读取。标准输入可与其他文件结合。

举例

在显示屏上输出 **myfile** 文件：

dos2ux myfile

将 **file1** 和 **file2** 转换为 DOS 格式，然后将其连接起来，放在 **file3** 中。

ux2dos file1 file2 > file3

返回值

dos2ux 和 **ux2dos** 成功时返回 **0** ，失败时返回 **2** 。唯一可能的失败原因是无法打开指定的文件，在这种情况下，命令会输出警告。

警告

类似以下格式的命令：

dos2ux file1 file2 > file1

会在连接开始前覆盖 **file1** 中的数据，从而导致 **file1** 内容的丢失。因此，使用 Shell 特殊字符时应谨慎小心。

另请参阅

doschmod(1)、 doscp(1)、 dosdf(1)、 dosls(1)、 dosmkdir(1)、 dosrm(1)、 dosif(4)。

名称

doschmod - 更改 DOS 文件的属性

概要

doschmod [**-mu**] *mode device***:** *file* ...

说明

doschmod 命令将从 HP-UX 中去除；请参阅下面的警告。

doschmod 是 **chmod** 的 DOS 对应命令（请参阅 *chmod*(1)）。

选项

doschmod 可识别一个选项：

-m 如果存在一个与卷标签同名的普通文件，则将对该文件而不是卷标签执行操作。

-u 禁用参数大小写转换。如果未指定该选项，则所有 DOS 文件名都会转换为大写字母。

DOS 文件名可由嵌入的冒号 (**:**) 分隔符来识别；有关 DOS 文件命名约定，请参阅 *dosif*(4)。

在指定 DOS 文件名时，可以使用元字符 *****、 **?** 和 **[...]**。由于必须通过 DOS 实用程序，而不是通过 Shell 来执行文件名扩展，因此在指定 DOS 文件名时，必须将这些符号用引号引起来。DOS 实用程序扩展文件名的方式如模式匹配 表示法中的 *regexp*(5) 中所述。

每个命名文件的属性会根据 *mode* 而改变，模式是一个介于 000 和 0377 之间的八进制数。 *mode* 是通过下列模式的逻辑 或构建的：

200	保留。请勿使用。
100	保留。请勿使用。
040	归档。每当写入并关闭文件时设置。
020	目录。请勿修改。
010	卷标签。请勿修改。
004	系统文件。标记属于 DOS 操作系统一部分的文件。
002	隐藏文件。标记使用 DOS DIR 命令时不会显示在 DOS 目录列表中的文件。
001	只读文件。将文件标记为只读。

警告

由于 **doschmod** 将从 HP-UX 中去除，因此最好不要使用该命令。

若 *mode* 的值指定不当，可能会导致文件和（或）目录无法被访问，在某些情况下还可能损坏文件系统。为防止发生这类问题，请不要更改目录的模式和卷标签。

普通用户应该无须使用除 001、002 和 040 之外的 *mode* 位。

举例

将文件 **/dev/rfd9122:memo.txt** 标记为隐藏文件：

doschmod 002 /dev/rfd9122:memo.txt

将文件 **driveC:autoexec.bat** 标记为只读：

doschmod 001 driveC:autoexec.bat

另请参阅

chmod(1)、 dos2ux(1)、 doscp(1)、 dosdf(1)、 dosls(1)、 dosmkdir(1)、 dosrm(1)、 chmod(2)、 dosif(4)。

名称

doscp - 复制到或从 DOS 文件复制

概要

doscp [**-fmvu**] *file1 file2*

doscp [**-fmvu**] *file1* [*file2* ...] *directory*

说明

doscp 命令将从 HP-UX 中删除；请参阅下面的警告。

doscp 是与 **cp** （请参阅 *cp*(1) ）对应的 DOS 命令。 **doscp** 可将 DOS 文件复制为 DOS 或 HP-UX 文件，将 HP-UX 文件复制为 HP-UX 或 DOS 文件，或者将 HP-UX 或 DOS 文件复制到 HP-UX 或 DOS 目录。参数列表中最后的名称是目标文件或目录。

DOS 文件名可由嵌入的冒号 (**:**) 分隔符来识别；有关 DOS 文件命名约定，请参阅 *dosif*(4) 。

在指定 HP-UX 和 DOS 文件名时，都可使用元字符 ***** 、 **?** 和 **[...]** 。由于必须通过 DOS 实用程序，而不是通过 Shell 来执行文件名扩展，因此在指定 DOS 文件名时，必须将这些符号用引号引起来。DOS 实用程序对文件名的扩展如 模式匹配 表示法下的 *regexp*(5) 中所述。

文件名 **-** （短线）被解释为标准输入或标准输出，这取决于它在参数列表中的位置。

选项

doscp 可识别下列选项：

- **-f** 无条件覆盖现有文件。如果没有使用这个选项，则 **doscp** 将请求允许覆盖现有 HP-UX 文件。
- **-v** 详细信息模式。 **doscp** 输出源文件名称。
- **-u** 禁用参数大小写转换。如果没有使用这个选项，则所有的 DOS 文件名都会转换为大写字母。
- **-m** 在这种情况下，可能会创建一个与 DOS 卷标签相同的文件名。

返回值

doscp 如果成功复制了所有文件则返回 0。否则，会将一条消息输出到标准错误中并返回非零值。

举例

将 HP-UX 目录 **abc** 中的文件复制到存储为 HP-UX 文件 **hard_disk** 的 DOS 卷：

```
doscp abc/* hard_disk:
```

将 DOS 文件 **/backup/log** 通过 HP-UX 专用文件 **/dev/rfd9127** 复制为位于当前目录下的 HP-UX 文件 **logcopy** ：

```
doscp /dev/rfd9127:/backup/log   logcopy
```

将存储为 HP-UX 文件 **bb** 的卷上的 DOS 文件 **zulu** 复制到标准输出：

```
doscp bb:zulu -
```

将 DOS 卷 **/dev/rdsk/c1t2d0** 上 **/dameron** 目录中的所有 **txt** 文件复制到位于当前目录中的 **abacus** HP-UX 目录下：

doscp '/dev/rdsk/c1t2d0:/dameron/*.txt' abacus

警告

建议不要使用 **doscp** ，因为该命令将从 HP-UX 中删除。可使用 *dos2ux*(1) 取代该命令。

对于 **doscp** ，使用原始设备专用文件 (**/dev/rdsk/**) 比使用块设备专用文件的可靠性更高。

要使用 SCSI 软盘设备，必须将设备驱动程序 **sflop** 配置到内核中（可以使用 **ioscan** 命令验证配置）。

另请参阅

cp(1)、 dos2ux(1)、 doschmod(1)、 dosdf(1)、 dosls(1)、 dosmkdir(1)、 dosrm(1)、 ioscan(1M) dosif(4)。

名称

dosdf - 报告可用磁盘群集的数目

概要

dosdf *device*[**:**]

说明

dosdf 命令将从 HP-UX 中去除；请参阅下面的警告。

dosdf 是 **df** 命令的 DOS 对应命令（请参阅 *df*(1) ）。它输出指定 DOS 卷上群集的大小（单位为字节）和可用群集的数目。

警告

最好不要使用 **dosdf** ，因为该命令将从 HP-UX 中去除。

另请参阅

df(1)、 dos2ux(1)、 doschmod(1)、 doscp(1)、 dosls(1)、 dosmkdir(1)、 dosrm(1)、 dosif(4)。

名称

dosls、dosll - 列出 DOS 目录的内容

概要

dosls [**-aAmudl**] *device*:[*file*] ...

dosll [**-aAmudl**] *device*:[*file*] ...

说明

dosls 和 **dosll** 命令将从 HP-UX 中去除；请参阅下面的警告。

dosls 是 **ls** 的 DOS 对应命令（请参阅 *ls*(1) ）。

对于每个命名的目录， **dosls** 将列出该目录的内容。对于每个命名的文件， **dosls** 将重复其名称以及所请求的其他任何信息。如果通过名称 **dosll** 名称调用，则暗含了 **-l** (ell) 选项的作用。

选项

dosls 和 **dosll** 可识别下列选项：

- **-a** 列出所有目录条目。如果没有该选项，则不列出隐藏的文件、系统文件以及其名称以点号 (**.**) 开头的文件。
- **-A** 与 **-a** 相同，例外的是不列出当前目录和父目录。对于超级用户，该选项缺省为已设置，并由 **-A** 禁用。
- **-m** 如果存在一个与卷标签同名的普通文件，则将对该文件而不是卷标签执行操作。
- **-u** 禁用参数大小写转换。如果不指定该选项，则所有 DOS 文件名都会转换为大写字母。
- **-d** 如果参数不是目录，仅列出其名称。通常与 **-l** 共同使用，以获取目录状态。
- **-l** 以长格式列出，给出每个文件的文件属性、字节数大小和上次修改时间，并列出 DOS 卷标。如果该选项与 **dosll** 命令一起使用，则禁用长格式列表。

DOS 文件名由嵌入的冒号 (**:**) 分隔符来识别；有关 DOS 文件命名约定，请参阅 *dosif*(4) 。

在指定 DOS 文件名时，可以使用元字符 ***** 、 **?** 和 **[...]** 。由于必须通过 DOS 实用程序，而不是通过 Shell 来执行文件名扩展，因此在指定 DOS 文件名时，必须将这些符号用引号引起来。DOS 实用程序扩展文件名的方式如模式匹配 表示法中的 *regexp*(5) 中所述。

警告

由于即将从 HP-UX 中删除 **dosls** 和 **dosll** 命令，因此最好不要使用这些命令。

举例

以下示例假设在通过 HP-UX 专用文件 **/dev/rdsk/c2t1d0** 访问的设备上存在 DOS 目录结构。

以下示例列出 DOS 目录结构的根目录中的所有文件：

dosls -a /dev/rdsk/c2t1d0:

以下示例列出 DOS 目录结构的根目录中带 **bat** 扩展名的所有文件：

dosls -a '/dev/rdsk/c2t1d0:*.bat'

以下示例生成有关 DOS 目录 **/dos/math** 的所有信息的长格式列表，但不列出该目录中的文件：

dosls -ld /dev/rdsk/c2t1d0:/dos/math

另请参阅

dos2ux(1)、 doschmod(1)、 doscp(1)、 dosdf(1)、 dosmkdir(1)、 dosrm(1)、 ls(1)、 dosif(4)。

名称

dosmkdir - 创建 DOS 目录

概要

dosmkdir [**-mu**] *device***:***directory* ...

说明

dosmkdir 命令将从 HP-UX 中去除；请参阅下面的警告。

dosmkdir 是 **mkdir** 命令的 DOS 对应命令（请参阅 *mkdir*(1)）。它创建指定的目录。标准条目（如表示目录本身的 **.** 及其父目录的 **..**）都是自动创建的。

可以使用一个选项：

-m 在这种情况下，可能会创建一个与 DOS 卷标相同的目录名。

-u 禁用参数大小写转换。如果没有使用这个选项，则所有的 DOS 文件名都会转换为大写字母。

DOS 文件名可由嵌入的冒号 (**:**) 分隔符来识别；有关 DOS 文件命名约定，请参阅 *dosif*(4)。

诊断信息

dosmkdir 如果成功创建了所有目录，则返回 0。否则，会将一条消息输出到标准错误中并返回非零值。

警告

最好不要使用 **dosmkdir**，因为该命令将从 HP-UX 中去除。

举例

在通过 HP-UX 专用文件 **/dev/rfd9122** 访问的设备上，在目录 **/math/lib** 下创建一个名为 **numbers** 的空子目录：

dosmkdir /dev/rfd9122:/math/lib/numbers

另请参阅

dos2ux(1)、 doschmod(1)、 doscp(1)、 dosdf(1)、 dosls(1)、 dosrm(1)、 mkdir(1)、 dosif(4)。

名称

dosrm、dosrmdir - 删除 DOS 文件或目录

概要

dosrm [**-fmriu**] *device*:*file* ...

dosrmdir [**-mu**] *device*:*file* ...

说明

dosrm 和 **dosrmdir** 命令将从 HP-UX 中去除；请参阅下面的警告。

dosrm 和 **dosrmdir** 是 **rm** 和 **rmdir** 的 DOS 对应命令（请分别参阅 *rm*(1) 和 *rmdir*(1) ）。

dosrm 从目录中删除一个或多个文件条目。如果指定的文件是一个目录，那么在未指定可选参数 **-r** 时将显示一条错误消息（请参阅下文）。

dosrmdir 在命名目录为空时将删除该条目。

选项

dosrm 和 **dosrmdir** 可识别下列选项:

- **-f** （强制）无条件地删除指定的文件（即使该文件被标记为只读）。
- **-r** 使 **dosrm** 以递归方式先删除某个目录中的所有内容，然后再删除该目录本身。 **dosrm** 能够以递归方式最多删除 17 级目录。
- **-i** （交互式）使 **dosrm** 询问是否删除每个文件。如果还指定了 **-r** ，则 **dosrm** 会询问是否检查遇到的每个目录。
- **-m** 如果存在一个与卷标签同名的普通文件，则将对该文件而不是卷标签执行操作。
- **-u** 禁用参数大小写转换。如果未指定该选项，则所有 DOS 文件名都会转换为大写字母。

DOS 文件名由嵌入的冒号 (**:**) 分隔符来识别；有关 DOS 文件命名约定，请参阅 *dosif*(4) 。

在指定 DOS 文件名时，可以使用元字符 ***** 、 **?** 和 **[...]** 。由于必须通过 DOS 实用程序，而不是通过 Shell 来执行文件名扩展，因此在指定 DOS 文件名时，必须将这些符号用引号引起来。DOS 实用程序扩展文件名的方式如模式匹配 表示法中的 *regexp*(5) 中所述。

警告

由于即将从 HP-UX 中去除 **dosrm** 和 **dosrmdir** 命令，因此最好不要使用这些命令。

举例

以下示例假设在通过 HP-UX 专用文件 **/dev/rfd9122**访问的设备上存在 DOS 目录结构。

以递归方式逐级搜索 DOS 目录 **/tmp** ，并询问是否强制删除每个 DOS 文件（即，不进行文件模式检查）：

dosrm -irf /dev/rfd9122:/tmp

从存储为 HP-UX 文件 **hard_disk** 的 DOS 卷中，删除 DOS 目录 **doug** ：

dosrmdir hard_disk:doug

另请参阅

dos2ux(1)、 doschmod(1)、 doscp(1)、 dosdf(1)、 dosls(1)、 dosmkdir(1)、 rm(1)、 rmdir(1)、 dosif(4)。

名称

du - 汇总磁盘利用率

概要

du [**-a**|**-s**] [**-bkrx**] [**-t** *type*] [*name* ...]

说明

du 命令可输出分配给由 *name* 操作数指定的每个目录和文件中所有文件和（递归式）目录的 512 字节块的数量。块计数包括文件间接占用的块。对于有两个或更多链接的文件只计算一次。如果未指定 *name* ，则使用当前的工作目录。

缺省情况下， **du** 只为 *name* 操作数和包含在这些分层结构中的每个目录生成一个统计条目。

选项

du 命令可识别下列选项：

-a 除正常输出外，还为目录分层结构中遇到的每个文件输出统计条目。

-b 如果 *name* 操作数表示一个目录，且已为其启用了文件系统交换，则会输出交换系统当前正在使用的块的数量。

-k 以 1024 字节块为单位进行块的计数。

-r 输出有关无法读取目录、无法访问文件等消息。 **du** 通常不会输出这些错误消息。

-s 只为每个指定的 *name* 操作数输出总磁盘利用率。

-x 只为与 *name* 操作数指定的文件具有相同设备的那些文件报告磁盘利用率。对于每个指定的 *name* 操作数下的整个目录分层结构，通常会报告其磁盘利用率。

-t *type* 只为具有指定 *type* 的文件系统报告磁盘利用率（ *type* 的示例值包括 **hfs** 、 **cdfs** 、 **nfs** 等）。可以指定多个 **-t** *type* 选项。对于每个指定的 *name* 操作数下的整个目录分层结构，通常会报告其磁盘利用率。

举例

显示当前工作目录及其中所有目录的磁盘利用率，并生成有关无法读取目录的错误消息：

du -r

显示除任何已挂接的 **cdfs** 或 **nfs** 文件系统以外的整个文件系统的磁盘利用率：

du -t hfs /

只显示根卷 (/) 上文件的磁盘利用率。不收集其他已挂接的文件系统的利用率统计信息：

du -x /

警告

对于包含空隙的文件的块计数结果不准确。

另请参阅

df(1M)、 bdf(1M)、 quot(1M)。

符合的标准

du：SVID2、SVID3、XPG2、XPG3、XPG4

名称

echo - 回显（输出）参数

概要

echo [*arg*] ...

说明

echo 在标准输出上输出以空白字符分隔、以换行符终止的多个参数。该命令也理解 C 类转义约定；注意不要与 Shell 中使用的 **** 冲突:

\a	输出警报字符
\b	退格
\c	输出行，不换行
\f	换页符
\n	换行符
\r	回车
\t	制表符
\v	纵向制表符
****	反斜杠
*****n*	ASCII 码为 1 位、2 位、3 位或 4 位八进制数 *n* 的 8 位字符，其第一个字符必须是零。
\0*num*	输出 8 位值，即零位、1 位、2 位或 3 位八进制数 *num*

echo 可用于生成命令文件中的诊断信息，并可将已知数据发送到管道。

注意

Berkeley **echo** 与此实现方式不同。前者不实现反斜杠转义符。但可使用 **-n** 选项获得 **\c** 转义符的语义。作为 **csh** 内置功能而实现的回显命令遵循 Berkeley 语义（请参阅 *csh*(1) ）。

外部语言环境影响

环境变量

LC_CTYPE 确定将 *arg* 解释为单字节和（或）多字节字符。

如果在环境中未指定 **LC_CTYPE** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省的“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量中包含无效设置，则 **echo** 认为所有国际化变量都被设为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

作者

echo 由 OSF 和 HP 开发。

另请参阅

sh(1)。

缺陷

第一个 **\c**后面的所有字符都不会输出。这一问题通常无关紧要。

符合的标准

echo：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

ed、red - 面向行的文本编辑器

概要

ed [**-p** *string*] [**-s**|**-**] [**-x**] [*file*]

red [**-p** *string*] [**-s**|**-**] [**-x**] [*file*]

说明

ed 命令执行面向行的文本编辑器。它通常用于脚本和非交互式编辑应用程序，这是因为即使可以通过交互方式使用该命令，但其他一些诸如 **vi** 和 **ex** 之类的编辑器在交互式环境中使用起来往往更加方便。

如果指定 *file* ， **ed** 将对命名的文件执行 **e** 命令（请参阅下文）；也就是说，将该文件读入 **ed** 的缓冲区以便进行编辑。

选项

可识别以下选项：

-p *string* 在命令模式下使用 *string* 作为提示字符串。缺省情况下没有提示字符串。

-s|**-** 禁止 **e** 、 **E** 、 **r** 和 **w** 命令输出字节计数，并禁止 **!** *command* 后的 **!** 提示。 **-** 选项已过时，会从将来的版本中删除。

-x 首先执行 **X** 命令来处理加密的文件。

文件处理

ed 对它所编辑的文件的副本执行操作；在给定 **w** （写入）命令之前，对副本所做的更改对原始文件无任何影响。所编辑的文本的副本驻留在称作 缓冲区的临时文件中。只有一个缓冲区。

red 是 **ed** 的受限版本，它仅允许编辑当前目录中的文件，并禁止通过 **!***shell-command* 执行 Shell 命令。如果试图绕过这些限制，则将导致错误消息 **restricted shell** 。

ed 和 **red** 均支持 *fspec*(4) 格式设置功能。在 **stty -tabs** 或 **stty tab3** 模式下包括格式指定作为 *file* 的第一行并通过控制终端调用 **ed** （请参阅 *stty*(1) 时，将在扫描 *file* 的过程中自动使用指定的制表位。例如，如果文件的第一行包含

<:t5,10,15 s72:>

制表位将设置在列 5、10 和 15，并将强制规定最大行长度为 72。

注意：当您输入文本时， **ed** 会展开制表符，这是因为每隔七列，它们将作为缺省值键入。

编辑器命令结构

ed 的命令具有简单的常规结构：零个、一个或两个 *addresses* ，后接单字符 *command* ，其后可能接有该命令的参数。这些地址指定缓冲区中的一行或多行。需要地址的每个命令都具有缺省地址，因此通常可以省略地址。

通常，一行上只能有一个命令。追加、更改和插入命令接受文本输入，然后文本输入将适当的情况下放入缓冲区。当 **ed** 接受追加、更改或插入命令之后的文本时，可以认为它是处于 输入模式。当处于输入模式时，将 不识别编辑器命令，而仅收集所有输入。要终止输入模式，请单独在一行的开头键入一个句点 (**.**) 。

正则表达式

ed 支持基本正则表达式 (RE) 语法（请参阅 *regexp*(5) ）以及以下附加内容:

- 空 RE（如 **//** ）等效于遇到的最后一个 RE。
- 如果 RE 或替换字符串的闭合分隔符（如 **/** ）将是换行符前的最后一个字符，则可省略该分隔符，这种情况下将输出地址行。以下命令对是等效的:

```
s/s1/s2      g/s1      ?s1
s/s1/s2/p    g/s1/p    ?s1?
```

行地址

要理解行地址，请记住 **ed** 包含指向 当前行的指针。一般而言，当前行是受命令影响的最后一行。当前行上给定命令的确切效果将在各个命名的说明部分进行讨论。地址按照以下规则进行解释:

1. 字符 **.** 引用当前行。
2. 字符 **$** 引用缓冲区的最后一行。
3. 十进制数字 *n* 引用缓冲区的第 *n* 行。
4. **'***x* 引用标有标记名字符 *x* 的行，该字符必须是小写字母。行标有下文所述的 **k** 命令。
5. 一对斜线内的 RE (**/***RE***/**) 引用通过从当前行后的行朝着缓冲区结尾向前搜索并在第一个包含匹配 RE 的字符串的行处停止而找到的第一个行。如有必要，搜索将在缓冲区的开头自动分行并继续进行到包括当前行，因此将搜索整个缓冲区。（另请参阅下面的警告）。
6. 一对问号内的 RE (**?***RE***?**) 引用通过从当前行前的行朝着缓冲区开头向后搜索并在第一个包含匹配 RE 的字符串的行处停止而找到的第一个行。如有必要，搜索将在缓冲区的末尾自动换，并继续进行包括当前行。（另请参阅下面的警告）。
7. 十进制数字后接加号 (**+**) 或减号 (**-**) 之后的地址指定加上或减去指定行数的地址。加号可以省略。
8. 如果地址以 **+** 或 **-** 开头，则相对于当前行进行加减计算。例如， **-5** 将被解释为 **.-5** 。
9. 如果地址以 **+** 或 **-** 结尾，则将分别在地址中加上或减去 1。作为该规则以及以上规则 8 的结果，地址 **-** 引用当前行之前的行。（要维持与编辑器早期版本的兼容性，在地址中遇到音调符 (**^**) 和 **-** 时，将对它们进行相同的解释）。此外，多个 **+** 和 **-** 字符的结尾具有累积效果，因此 **--** 引用当前行之前的第二行。
10. 为方便起见，逗点 (**,**) 表示地址对 **1,$** ，而分号 (**;**) 表示 **.,$** 对。

命令需要零个、一个或两个地址。不使用地址的命令会将地址的出现当作错误进行处理。接受一个或两个地址的命令将在指定的地址数不足时使用缺省地址。如果指定的地址多于给定命令所需的地址，则将根据情况使用最后一个或两个命令。

地址通常使用逗点 (**,**) 相互分隔。它们也可以使用分号 (**;**) 进行分隔，这种情况下当前行 (**.**) 设置为第一个地址，并在其后计算第二个地址。该功能可用于确定向前和向后搜索的起始行（请参阅上面的规则 5 和 6）。任何双地址序列的第二个地址必须对应于缓冲区中与第一个地址对应的行之后的行。

编辑器命令

在以下 **ed** 命令列表中，缺省地址显示在括号中（括号不是地址的一部分，除非用于其他目的，否则不应放在实际的命令中）。

一行上出现多个命令通常是非法的。但是，任何命令（除 **e** 、 **f** 、 **r** 或 **w** 之外）均可带有 **l** 、 **n** 或 **p** 后缀，在这种情况下，当前行将分别按照 **l** 、 **n** 和 **p** 命令所述进行列表、编号或输出。

(.)a
text
.

a （追加）命令读取 *text* ，并将其追加在地址行之后。完成时，新的当前行是插入的最后一行，如果没有添加文本，则是地址行。地址 0 对于该命令是合法的，它使得追加的 *text* 放置在缓冲区的开头。

(.,.)c
text
.

c （更改）命令删除地址行，然后输入文本以替换删除的行。完成时，新的当前行是 *text* 中的最后一行，如果没有提供文本，则是删除的行之后的第一行。

(.,.)d

d （删除）命令从缓冲区中删除地址行。完成时，新的当前行是所删除文本之后的第一行，如果删除的行位于缓冲区的末尾，则是文件中的最后一行。

e *file*

e （编辑）命令删除缓冲区的全部内容，然后读入命名的 *file* 。完成时，新的当前行是缓冲区中的最后一行。如果没有给定文件名，则将使用记忆的文件名（请参阅 **f** 命令）。将显示读取的字符数，并记忆 *file* ，它在后继的 **e** 、 **r** 或 **w** 命令中可能会用作缺省的文件名。

如果 *file* 名以 **!** 开头，则行的其余部分将被解释为将读取其标准输出的 Shell 命令。这样的 Shell 命令将不会记忆为当前文件名。

另请参阅下面的诊断信息。

E *file*

E （强制编辑）命令等效于 **e** ，例外的是它不检查并确保自上一 **w** 命令以来未更改当前缓冲区。

f *file*

如果指定了 *file* ， **f** （文件名）命令会将记忆的文件名更改为 *file* 。否则，它将输出记忆的文件名。

(1,$)g/*RE***/***command-list*

g （全局）命令首先标记与给定 RE 匹配的每一行。这样，对于每个这样的行，将在当前行最初设置为该行的情况下执行 *command-list* 。单个命令或者命令列表中的第一个命令将与全局命令出现在同一行上。多行列表中，除最后一行之外的所有行必须以反斜杠 (\) 结尾。允许使用 **a** 、 **i** 和 **c** 命令以及关联的输入。如果 **.** （它通常终止输入模式）是 *command-list* 的最后一行，则可以将其省略。空的 *command-list* 等效于 **p** 命令。 **g** 、 **G** 、 **v** 和 **V** 命令不能在 *command-list* 中使用（另请参阅下面的警告）。

(1,$)G/*RE***/**

交互式 **G** （全局）命令首先标记与给定 RE 匹配的每一行。因此，对于每个这样的行，将输出该行，然后将当前行更改为该行，并可以输入和执行一个命令（ **a** 、 **c** 、 **i** 、 **g** 、 **G** 、 **v** 或 **V** 除外）。执行该命令后，将输出下一个带标记的行，依此类推。换行符充当空命令， **&** 将导致在当前 **G** 调用中重新执行最新的命令。请注意，作为 **G** 命令执行一部分输入的命令可能会影响缓冲区中的 任何行。 **G** 命令可以通过中断信号（ASCII DEL 或 BREAK）终止。

h — **h** （帮助）命令将提供一条简单的错误消息，解释最新的 **?** 诊断信息的原因。

H — **H** （帮助）命令将使得 **ed** 进入如下模式：输出所有后继 **?** 诊断消息的错误消息。它还会解释先前的 **?** （如果有）。 **H** 命令可打开或关闭该模式。该模式最初会关闭。

(.)i
text
.
— **i** （插入）命令在地址行之前插入给定的 *text* 。完成时，当前行是插入的最后一行，如果没有，则是地址行。该命令与 **a** 命令的唯一区别在于输入文本的放置。地址 0 对于该命令是非法的。

(.,.+1)j — **j** （连接）命令通过删除适当的换行符来连接相邻的行。如果正好给定了一个地址，该命令将不执行任何操作。

(.)k*x* — **k** （标记）命令用名称 *x* （必须是小写字母）标记地址行。然后，地址 **'***x* 将确定该行的地址。完成时，新的当前行保持不变。

(.,.)l — **l** （列表）命令以在视觉上非常明确的方式将地址行写入标准输出。下表列出的字符作为对应的转义序列写入。不在表中的非打印字符作为三位八进制数字（表示字符中的每个字节，最重要的字节优先）写入（带有前导的反斜杠字符）。

较长的行通过折叠点进行折叠，折叠点由反斜杠后接换行符来表示。每行的末尾标有 **$** 。 **l** (ell) 命令可以追加到 **e** 、 **E** 、 **f** 、 **q** 、 **Q** 、 **r** 、 **w** 或 **!** 之外的任何命令。当前行号将设置为写入的最后一行的地址。

转义序列	表示	**ASCII** 名称	转义序列	表示	**ASCII** 名称
****	反斜杠	\	**\r**	回车	CR
\a	警报	BEL	**\t**	横向制表符	HT
\b	退格	BS	**\v**	纵向制表符	VT
\f	换页符	FF			

(.,.)m*a* — **m** （移动）命令将地址行重定位到通过 *a* 指定地址的行之后。地址 0 对于 *a* 是合法的，这将使得地址行移动到文件的开头。如果地址 *a* 位于移动行的范围内，这就是错误；完成时，新的当前行是移动的最后一行。

(.,.)n — **n** （编号）命令输出地址行，并在每一行前添加其行号和一个制表符。完成时，新的当前行是输出的最后一行。 **n** 命令可以追加到 **e** 、 **f** 、 **r** 或 **w** 之外的任何命令。

(.,.)p — **p** （输出）命令输出地址行。完成时，新的当前行是输出的最后一行。 **p** 命令可以追加到 **e** 、 **E** 、 **f** 、 **q** 、 **Q** 、 **r** 、 **w** 或 **!** 之外的任何命令。例如， **dp** 删除当前行并输出新的当前行。

P — **P** （提示）命令使得 **ed** 用星号 **(*)** （如果在命令行中指定 **-p** 选项，则使用 *string* ）提示所有后继的命令。 **P** 命令可打开或关闭该模式。如果指定了 **-p** 选项，它最初会打开；否则，该模式将关闭。当前行号保持不变。

q — **q** （退出）命令使得 **ed** 退出。不执行文件自动写入（但请参阅下面的诊断信息）。

Q 编辑器无条件退出，而不检查自上一个 **w** 命令后缓冲区中的更改。

($)r 文件 **r**（读取）命令将指定的 *file* 读入缓冲区中的地址行后。如果没有给定文件名，则将使用记忆的文件名（请参阅 **e** 和 **f** 命令）。除非 *file* 是自调用 **ed** 以来使用的第一个文件名，否则记忆的文件名将保持不变。地址 0 对于 **r** 是合法的，并将 *file* 的内容放在缓冲区的开头。如果读取成功，则显示所读取的字符数。完成时，新的当前行是读入缓冲区中的最后一行。如果 *file* 名以 **!** 开头，则行的其余部分将被解释为将读取其标准输出的 Shell 命令。例如， **$r !ls** 将当前目录中文件的列表追加到所编辑文件的结尾。 Shell 命令将不会记忆为当前文件名。

(.,.)s/*RE***/***replacement***/***flags*

s（替换）命令搜索指定 RE 的实例的每个地址行。在找到匹配的每个行中，如果全局替换指示符 **g** 出现在命令后，所有（非重叠的）匹配字符串均替换为 *replacement* 。如果全局指示符未出现，则仅替换匹配字符串的第一个实例。如果数字 *n* 出现在命令后，则仅替换每个地址行上匹配字符串的第 *n* 个实例。如果替换在所有地址行上失败，这就是错误。可以使用除空格或换行符之外的任意字符替代 **/** 来分隔 RE 和 *replacement* 。完成时，新的当前行是进行替换的最后一行。（另请参阅下面的警告）。

如果 & 号 (**&**) 出现在 *replacement* 中，它将替换为当前行上匹配 RE 的字符串。 **&** 在该环境中的特殊含义可以通过在它前面添加 **** 来禁止。

作为更加常用的功能，字符 *****n*（其中 *n* 是一个数位）将替换为 **\(** 和 **\)** 之间的指定 RE 第 *n* 个正则子表达式所匹配的文本。当存在带括号的嵌套子表达式时， *n* 通过从左侧开始计算 **\(** 出现的次数来确定。

当字符 **%** 是 *replacement* 中的唯一字符时，最近的替换命令中使用的 *replacement* 将用作当前替换命令中的 *replacement* 。如果 **%** 所在的替换字符串包含多个字符或者前面带有 **** ，则将丢失其特殊含义。

一个行可以通过在其中替换换行符来进行拆分。 *replacement* 中的换行符必须通过在前面添加 **** 来进行转义。这种替换不能作为 **g** 或 **v** 命令列表的一部分来执行。

flags 的值是零个或更多个：

n 仅替换在每个地址行上找到的 RE 的第 *n* 个实例。

g 替换每个地址行上的 RE 的所有非重叠实例。

l 将最终的行写入在其中进行替换的标准输出。该行按照为 **l** 命令指定的格式写入。

n 将最终的行写入在其中进行替换的标准输出。该行按照为 **n** 命令指定的格式写入。

p 将最终的行写入在其中进行替换的标准输出。该行按照为 **p** 命令指定的格式写入。

(.,.)t*a* 与 **m** 命令相同，例外的是地址行的副本将放置在地址 *a*（可以是 0）之后。完成时，新的当前行是副本的最后一行。

u **u**（撤消）命令使得修改缓冲区中任意内容的最近命令（即最近的 **a** 、 **c** 、 **d** 、 **g** 、 **G** 、 **i** 、 **j** 、 **m** 、 **r** 、 **s** 、 **t** 、 **v** 或 **V** 命令）无效。如果通过 **g** 、 **G** 、 **v** 或 **V** 全局命令对缓冲区做出的所有更改将作为单一更改"撤消"；如果没有通过这些全局命令做出更改（例如使用 **g/***RE***/p**），则 **u** 命令将不起作用。当前行号设置为在命令开始后立即获得的值。

(1,$)v/*RE***/***command-list*

全局命令 **g** 的补充，即在第一步中标记的行是 不匹配 RE 的行。

(1,$)V/*RE***/** 交互式全局命令 **G** 的补充，即在第一步中标记的行是 不匹配 RE 的行。

(1,$)w 文件 **w**（写入）命令将地址行写入命名的文件。如果不存在该文件，除非当前 **umask** 设置另行指定，则将使用模式 666（任何用户均可读写）创建该文件（请参阅 *umask*(1)）。除非 *file* 是自调用 **ed** 以来遇到的第一个文件名，否则记忆的文件名将保持不变。如果没有给定文件名，则将使用记忆的文件名（请参阅 **e** 和 **f** 命令）。完成时，当前行地址保持不变。如果该命令成功，则显示所写入的字符数。

如果 *file* 名以 **!** 开头，则行的其余部分将被解释为其标准输入是标准行的 Shell 命令。这样的 Shell 命令将不会记忆为当前文件名。

X 密钥字符串从标准输入中请求。后继的 **e** 、 **r** 和 **w** 命令将使用 *crypt*(1) 的算法来通过该密钥加密和解密文本。显式的空密钥将关闭加密。

($)= 将显示地址行的行号。该命令不会更改当前行地址。

!*shell-command* 行中 **!** 后的剩余部分将发送到 Shell 进行解释并作为命令执行。在该命令的文本中，未转义的字符 **%** 将替换为记忆的文件名。如果 **!** 作为 Shell 命令的第一个字符出现，它将被替换为上一个 Shell 命令的文本。因此， **!!** 将重复上一个 Shell 命令。如果执行任何扩展，则将回显扩展的行。完成时，当前行地址保持不变。

(.+1) 换行符 行上的地址本身即可致使输出该地址行。换行符本身等效于 **.+1p** 。该技术对于在缓冲区中向前步进非常有用。

如果发送了中断符号（ASCII DEL 或 BREAK）， **ed** 将输出 **?** 并返回其命令级别。

存在以下大小限制：每个全局命令列表 256 个字符，每个文件名 64 个字符，缓冲区中 32 MB 字符。对行数的限制取决于用户内存量：每行包含 1 个单词。

外部语言环境影响

环境变量

SHELL 确定在所有 **!** 式命令中使用的首选命令行解释程序。如果该变量为空或者未设置，则使用 POSIX Shell **/usr/bin/sh**，（请参阅 *sh-posix*(1)）。

如果已设置， **TMPDIR** 指定用于临时文件的目录，它覆盖缺省目录 **/tmp** 。

LANG 为没有设置或为空的国际化变量提供了缺省值。如果 **LANG** 未设置或为空，则会使用缺省值"C"（请参阅 *lang*(5)）。如果任一国际化变量中包含无效设置，则所有国际化变量都会缺省为"C"。请参阅 *environ*(5)。

如果 **LC_ALL** 设置为非空的字符串值，它将覆盖所有其他国际化变量的值，包括 **LANG** 。

LC_CTYPE 确定文本作为单字节和（或）多字节字符的解释、字符作为可打印字符的分类以及在常规表达式中按字符类表达式匹配的字符。

LC_MESSAGES 确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息以及写入到标准输出的信息消息的格式和内容。

NLSPATH 确定消息目录的位置，以便处理 **LC_MESSAGES** 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

? 命令错误。使用 **h** 或 **H** 获取详细的解释。

? *file* 无法访问的文件。使用 **h** 或 **H** 获取详细的解释。

如果自上一个写入整个缓冲区的 **w** 命令以来在缓冲区中进行了更改， **ed** 将在您尝试用 **e** 或 **q** 命令毁坏缓冲区时发出警告。 **ed** 显示 **?** 或 **warning: expecting 'w'** ，然后继续进行常规的编辑，直到您输入另一个 **e** 或 **q** 命令，此时将执行该命令。 **-s** 或 **-** 命令行选项禁止该功能。

举例

通过 Shell 脚本在 **file-1** 中进行简单的替换，将任何行中 **abc** 的第一个实例替换为 **xyz** ，然后将更改保存到 **file-2** 中。

```
cat - << EOF | ed -s file-1
1,$ s/abc/xyz/
w file-2
q
EOF
```

请注意，如果命令失败，编辑器将立即退出。

警告

ed(1) 允许 **Maximum Line Length** 为 **4096** 个字符。如果试图创建长于该允许限制的行，则将使得 **ed(1)** 生成 **Line too long** 错误消息。

! 命令不能服从于 **g** 或 **v** 命令。

如果从受限的 Shell 中调用编辑器，则不能使用 **!** 命令以及从 **e** 、 **r** 和 **w** 命令中转义的 **!**

（请参阅 *sh*(1) ）。

正则表达式中的序列 **\n** 不匹配换行符。

l 命令无法正确处理 DEL。

不能编辑直接使用 **crypt** 命令通过空密钥加密的文件（请参阅 *crypt*(1) ）。

如果编辑器输入来自命令行（例如 **ed file < ed-cmd-file)**），则编辑器在命令文件中命令第一次失败时退出。

读取文件时， **ed** 将忽略 ASCII NUL 字符以及最后一个换行符后的所有字符。当使用正则表达式搜索包含 NUL 字符的字符序列或文件结束标志附近的文本时，这可能导致意外的行为。

作者

ed 由 HP 和 OSF 联合开发。

文件

/tmp/e*p* 其中 *p* 为进程号的临时缓冲区文件。

ed.hup 如果终端挂起，所做工作将保存在此处。

另请参阅

awk(1)、 csh(1)、 crypt(1)、 ex(1)、 grep(1)、 ksh(1)、 sed(1)、 sh(1)、 sh-posix(1)、 stty(1)、 vi(1)、 fspec(4)、 environ(5)、 lang(5)、 regexp(5)。

《Text Processing: User's Guide》中的 **ed** 部分。

符合的标准

ed：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

red：SVID2、SVID3、XPG2、XPG3

名称

elfdump - 转储在对象文件中包含的信息。

概要

elfdump [**-acCdfghHjkLopqrsStuUvV**] [**-dc**] [**-dl**] [**-tx**] [**-tv**] [**-D** *num*] [**+D** *num2*] [**+interp**] [**+linkmap**] [**+linkmap_bss**] [**+linkmap_file**] [**-n** *name*] [**+objdebug**] [**+s** *section*] [**-T** *num*] [**+T** *num2*] *files*...

说明

elfdump 接收一个或多个对象文件或库，并转储有关它们的信息。支持以下选项:

-a 从归档库转储归档标题。

-c 转储字符串表。

-C （修饰符）在输出 C++ 符号名称之前破坏它们。此修饰符对 **-c** 、 **-r** 、 **-s** 和 **-t** 有效。如果与 **-H** 一起指定，则忽略此修饰符。如果与 **-n** *name* 一起指定，则与 *name* 匹配的未破坏名称的符号将被输出，并且其符号名称将输出为已破坏的名称。

-d 输出包含编译单元词典和链接程序占用量的 .note 节。此选项与 **elfdump -dc -dl** 具有相同的效果。

-dc 输出 .notes 节的编译单元词典。

-dl 输出 .notes 节的链接程序占用量。链接程序占用量具有有关用于生成文件的链接程序以及链接时间的信息。

-D *num* （修饰符）输出其索引为 *num* 的节。

+D *num2* （修饰符）输出 1 到 *num2* 的范围内的节。如果与 **-D** 一起使用，则将输出 *num* 到 *num2* 的范围内的节。对 **-h** 、 **-r** 、 **-s** 有效。如果与 **-r** 一起使用，则仅输出适用于范围内节的重定位。

-f 转储文件标题（ELF 标题）。

-g 从归档转储全局符号。

-h 转储节标题。

-H （修饰符）对所有选项均以十六进制、八进制或十进制格式转储输出信息。

+interp 显示 **a.out** 的运行时解释程序路径名（通常为动态加载程序和微加载程序的位置）。只有共享的绑定可执行程序才具有此字符串。要更改该设置，请使用 **ld +interp** 命令。

-j 如果源文件已使用 **+objdebug** 选项进行编译或与 **+tools** （仅限 PA-RISC）选项链接在一起，则输出一个或多个可执行文件的对象词典。对象词典条目包含构成特定节的对象文件的名称、节内的相对偏移量、对象文件作用的大小和条目的属性。

-k 根据词典成员关系输出 CTTI 节标题。

-L 转储共享库和动态链接的程序文件中的 **.dynamic** 节。

+linkmap 输出 **.linkmap** 节；仅当使用增量链接程序（通过 **ld +ild** 命令）或使用链接程序选项 **+objdebug**（它是缺省值）以及编译程序选项 **-g** （它不是缺省值）时才创建该节。

+linkmap_bss 输出 **.linkmap_bss** 节；仅当使用增量链接程序（通过 **ld +ild** 命令）或使用链接程序选项 **+objdebug** （它是缺省值）以及编译程序选项 **-g** （它不是缺省值）时才创建该节。

+linkmap_file 输出 **.linkmap_file** 节；仅当使用增量链接程序（通过 **ld +ild** 命令）或使用链接程序选项 **+objdebug** （它是缺省值）以及编译程序选项 **-g** （它不是缺省值）时才创建该节。

-n *name* （修饰符）转储有关指定节或符号 *name* 的信息。此选项对 **-h** 、 **-r** 、 **-s** 和 **-t** 有效。如果与 **-t** 一起使用，则 *name* 与符号名称有关，并且 **elfdump** 将仅转储其名称与 *name* 匹配的符号条目。如果与其他选项一起使用，则 *name* 与节名有关，并且 **elfdump** 将仅转储其名称与它匹配的节。

-o 转储可选标题（程序标题）。

-p （修饰符）对于所有选项都不输出标题。

-q （修饰符）禁止输出 CTTI 节标题。对 **-k** 选项有效。

-r 转储重定位。

-s 转储节内容。

+objdebug 将以 **.objdebug_** 开头的任何节转储为字符串表。

+s *name* （修饰符）转储由 *name* 指定的节。仅对 **-c** 和 **-t** 有效。

-S （修饰符）以短格式转储输出信息。对 **-h** 和 **-o** 选项有效。

-t 转储符号表条目。

-tx （基于 Itanium 的系统）除了通过 **-t** 选项转储的信息外，还转储符号表中 **st_shndx** 的值。此选项对于验证在符号表中存储的数据是很有用的。

-T *num* 输出其索引为 *num* 的符号。

+T *num2* （修饰符）输出 0 到 *num2* 的范围内的符号。如果与 **-T** 一起使用，则输出 *num* 到 *num2* 的范围内的符号。对 **-t** 有效。

-tv 输出版本化符号。

-u 输出用法菜单。

-U 输出展开表。

-v （修饰符）在输出之前验证 CTTI 节标题。对 **-k** 选项有效。

-V 输出 **elfdump** 的版本号。

举例

查看从共享库导出的函数：

$ **elfdump -s -n .dynsym libsubs.so | grep 'FUNC GLOB' | grep -v UNDEF**

查看从共享库导出的全局数据项：

$ **elfdump -s -n .dynsym libsubs.so | grep 'OBJT GLOB' | grep -v UNDEF**

显示字符串表信息 (**.strtab**)：

$ **elfdump -c subs.o**

列出与程序或共享库（相关库）链接在一起的共享库：

$ **elfdump -L a.out | grep Needed**
$ **chatr a.out** # shared library list

列出由程序打开的共享库的嵌入式路径：

$ **elfdump -L a.out | grep Rpath** # or
$ **elfdump -s -n .dynamic a.out | grep Rpath**
$ **chatr a.out** # embedded path

另请参阅

系统工具

ld(1) 调用链接编辑器

其他信息

a.out(4) 汇编程序、编译程序和链接程序输出
elf(3E) 可执行程序和链接格式

文本和教程

《HP-UX Linker and Libraries Online User Guide》
（请参阅 **+help** 选项）
《HP-UX Linker and Libraries User's Guide》
（有关订购信息，请参阅 *manuals*(5) ）
《HP-UX Software Transition Toolkit (STK) -- ELF Object Formats》
http://www.software.hp.com/STK/

名称

elm - 通过面向屏幕的界面处理电子邮件

概要

elm [**-aKkmtVz**] [**-f** *folder*]

elm [**-s** *subject*] *address-list*

elm -c [*alias-list*]

elm -h

elm -v

说明

elm 程序是面向屏幕的电子邮件处理系统。它支持业内广泛使用的 MIME 标准（针对非文本邮件）、特殊的表单消息和表单回复机制，并为个人和工作组提供简便易用的别名系统。 **elm** 有三种主要的操作模式：

- 交互模式，即作为交互式邮件界面程序运行（第一条语法）。
- 消息模式，即通过 Shell 命令行将单个交互式消息发送到邮件地址列表。（第二条语法）。
- 文件模式，即通过命令行管道或重定向将文件或命令输出发送到邮件地址列表（第二条语法）。

在所有三种情况下， **elm** 接受在 **elm** 别名数据库和系统 **elm** 别名数据库中的 **elmrc** 初始化文件中设置的值。

下面将按相反的顺序介绍这些模式（从最短的说明到最长的说明）。

选项

可识别以下选项：

-a 设置 **arrow=ON** 。使用箭头 (**->**) 而不是反白条来标记各个索引中的当前项目。这将覆盖 **arrow** 布尔变量的设置（请参阅 ELM 配置部分）。

-c 检查别名。对照个人 **elm** 别名数据库和系统 **elm** 别名数据库检查 *alias-list* 中的别名。结果将写入标准的输出。首先会报告错误，其格式如下：

> **(**别名 **"***alias***"** 未知**)**

如果以标头条目的格式报告成功消息，并且组别名替换为它们的成员，则其格式如下：

> 扩展为**:** *alias-address* **(***fullname***),**
> *alias-address* **(***fullname***),**
> ...
> *alias-address* **(***fullname***)**

如果没有 *fullname* ，则将忽略“ **(***fullname***)**”部分。

-f *folder* 文件夹文件。从 *folder* 文件而不是收件箱中读取邮件。文件夹文件采用标准的邮件文件格式，即由邮件系统创建的格式或由 **elm** 本身保存的格式。

-h 帮助。显示命令行选项的注释列表。

-k 设置 **softkeys=OFF** 。禁用功能键（HP 2622 功能键）。这将覆盖 **softkeys** 布尔变量的设置（请参阅 ELM 配置部分）。

-K 设置 **keypad=OFF** 和 **softkeys=OFF** 。禁用功能键和箭头光标键。如果您的终端没有 HP 2622 功能键协议，该选项就是必需的。这将覆盖 **keypad** 和 **softkeys** 布尔变量的设置（请参阅 ELM 配置部分）。

-m 设置 **menu=OFF** 。在几个“交互模式”屏幕上不显示命令菜单。这将覆盖 **menu** 布尔变量的设置（请参阅 ELM 配置部分）。

-s *subject* 主题。指定“文件模式”或“消息模式”消息的主题。

-t 设置 **usetite=OFF** 。不使用 **termcap ti/te** 和 **terminfo cup** 光标定位条目。这将覆盖 **usetite** 布尔变量的设置（请参阅 ELM 配置部分）。

-V 冗余传输。使用 **-v** 选项将外发的消息发送到 **sendmail** 邮件传输代理（请参阅 *sendmail*(1M) ）。

-v 版本。输出 **elm** 版本信息。它显示版本号和指定或省略的编译信息。

-z 零。如果收件箱中没有邮件，则不进入 **elm** 。

操作数

可识别以下操作数:

address-list 由空白字符分隔的一个或多个邮件地址列表、 **elm** 用户别名列表或 **elm** 系统别名系统别名列表。

alias-list 由空白字符分隔的一个或多个 **elm** 用户别名列表或 **elm** 系统别名列表。

术语

本联机帮助页中将使用以下术语。

“空白字符”
空格或制表符，有时称作换行空白符 (linear white space)。

“正文” 消息的正文。请参阅 消息。

“布尔变量”
请参阅 “配置变量”。

“配置变量”
布尔、数值或字符串变量，定义 **elm** 邮件系统中的缺省行为。请参阅 ELM 配置部分。

“**elm** 系统别名文本文件”
: **elm** 系统别名数据库的源文件 **/var/mail/.elm/aliases.text** 。

“**elm** 用户别名文本文件”
: 用户自己的 **elm** 别名数据库的源文件 **$HOME/.elm/aliases.text** 。

“**elm** 用户标头文件”
: **$HOME/.elm/elmheaders** 文件，用户可以在其中指定特殊的标头条目，这些标头条目将包括在所有外发的消息中。

“**elmrc** 配置文件”
: **$HOME/.elm/elmrc** 文件，定义 **elm** 配置变量的初始值。

“环境变量”
: 在 Shell 中设置的用于调用 **elm** 的全局变量。请参阅“外部语言环境影响”部分。

“文件夹”
: 包含邮件消息的文件，这些邮件采用 **sendmail** 或 **elm** 创建的格式。

“完整名称”
: 用户的名字和姓氏，从别名文本文件或 **/etc/passwd** 文件中提取。

“标头” 消息的标头。请参阅 消息。

“标头条目”
: 消息标头部分中的条目，有时称作标头字段。

“收件箱”
: 接收邮件的邮箱，通常是 **/var/mail/***loginname* 。

“邮件目录”
: 由 **maildir** 字符串变量定义的目录，用户通常将邮件消息存储在其中的文件夹中。

“邮件传输代理 **(MTA)**”
: 与其他系统进行邮件消息收发的程序。在 HP-UX 系统上，MTA 是 **sendmail** （请参阅 *sendmail*(1M)）。

“**mailcap**”
: 该文件包含有关如何撰写和显示不仅仅是七或八位 ASCII 字符的邮件消息的信息。

“**metamail**”
: 处理非文本邮件消息的系统程序。

“消息” 文件夹中由消息分隔符、标头和正文组成的文本行的序列。消息分隔符为如下所示的行形式：

From *sender date*

标头在消息分隔符后开始，以第一个空行结束。 正文在空行处开始，在下一个消息分隔符处结束。正文可以包括小节，称作 附件或 “正文部分” ，它们由边界分隔符、标头和正文组成。该过程可以是

递归的。有关详细信息，请参阅“METAMAIL 配置”部分。

“数值变量”
请参阅 “配置变量”。

“**sendmail** 别名数据库”
由 **sendmail** MTA 用来定向本地邮件的别名数据库 **/etc/mail/aliases** 。

“签名文件”
追加到外发消息的文件，通常包含有关用户自身的信息。可以有两个签名文件，一个用于发送到本地计算机的消息，另一个用于其他消息。请参阅 **localsignature** 和 **remotesignature** 字符串变量。

“字符串变量”
请参阅 “配置变量”。

“用户名”
通常是邮件收件人的登录名或邮箱名。

“变量” 请参阅 “配置变量”和 “环境变量”。

文件模式

如果标准输入连接到管道或文件，并且指定了 *address-list* ，则 **elm** 将以“文件模式”运行。

管道中前一命令的输出或者该文件的内容将通过邮件发送到 *address-list* 的成员。 *address-list* 将展开，它基于您的 **elm** 别名数据库和系统 **elm** 别名数据库，并放置在 收件人**:** 标头条目中。

如果省略 **-s** 或者 *subject* 为空，则 *subject* 将缺省为:

邮件无主题 **(file transmission)**

subject 的表达值或缺省值放置在 主题**:** 标头条目中。

请参阅“举例”部分。

消息模式

如果标准输入连接到终端，并且指定了 *address-list* ，则 **elm** 将以“消息模式”运行。

address-list 将展开，它基于您的 **elm** 别名数据库和系统 **elm** 别名数据库，并放置在“ 收件人**:** ”标头条目中。“ 收件人**:** ”标头条目的显示格式与“交互模式”中“消息菜单” **m** （邮件）命令的格式相同。

如果 *subject* 的值是非空或者是空字符串，则将其放置在 主题**:** 标头条目中，并将显示 主题**:** 行供修改。

如果 **askcc** 在 **elmrc** 文件中是 **ON** ，则将提示您使用 抄送**:** 。

然后将启动 **editor** 字符串变量（如果未添加签名文件）或者 **alteditor** 字符串变量（如果已添加签名文件）定义的编辑器，以便您编写消息。

退出编辑器后，将进入“发送菜单”，如“交互模式”部分所述。

如果选择“发送菜单” **s** （发送）命令，则将发送消息，并且程序将终止。如果选择“发送菜单” **f** （忘记）命

令，则会将消息存储在 **$HOME/Canceled.mail** 中，并且程序将终止。如果选择其他命令，则将执行相应的操作。

请参阅“举例”部分。

交互模式

如果标准输入连接到终端，并且未指定 *address-list* ，则 **elm** 将以面向屏幕的“交互模式”运行。

如果没有 **$HOME/.elm** 目录，或者没有由 **maildir** 字符串变量定义的邮件目录，系统将依次询问是否要将其创建。可以回答 **y** 表示 是， **n** 表示 否，或者 **q** 表示 退出。对于 **y** 或 **n** ，将在适当的情况下创建或不创建这些目录，并且程序将继续。对于 **q** ，程序将终止。

概述

在调用时， **elm** 将从文件 **$HOME/.elm/elmrc** （如果有）中读取自定义变量来初始化参数。该文件可以在 **elm** 中进行保存，而其中的一些变量也可使用“消息菜单” **o** （选项）命令进行修改。

elm 首先显示“主菜单”或“消息菜单”，该菜单显示收件箱或选定邮件文件夹中的消息的索引条目。利用其他选项，可以阅读、打印、回复和转发这些消息，也可以开始编写要发给其他用户的新邮件消息。

也可以转到“别名菜单”，在其中可以创建、修改和删除您的个人别名。通过“别名菜单”，可以选择一个或多个别名并向相应的用户发送消息。

发送消息时，可以包含多种格式的附件，如 PostScript、图像、音频、视频和纯文本。附件可以单独管理，这对于您和您的通讯者都比较方便。

发送消息

发送消息时，将使用由 **editor** 或 **alteditor** 字符串变量定义的编辑器。如果编辑器为 **builtin** ，则在编写消息时，可使用“内置编辑器”小节所述的命令集。

如果存在 **elmheaders** 文件（请参阅标头文件部分），该文件中的所有非空白行均将复制到所有外发邮件的标头。这对于添加特殊信息标头（如 **X-Organization:** 、 **X-Phone:** 等）非常有用。

MIME 支持

对于标头和消息， **elm** 支持 MIME 协议（RFC 1521 和 RFC 1522），从而可用于查看和发送包含普通 ASCII 文本之外的内容的邮件。例如，邮件内容可以是音频、视频、图像等，也可以是这些内容的组合。

它还支持使用 MIME 编码 （**base64** 和 **quoted-printable** ）将 8 位数据转换为 7 位，从而可符合 SMTP (RFC 821)，该协议只允许在消息中使用 7 位字符。

elm 还提供了用于查看多部分 MIME 消息的工具。如果 **elm** 收到其类型不是 **text/plain** 的消息，它将调用 **metamail** ，后者调用适当的实用程序（例如 **ghostview** 、 **xv** 、音频编辑器和 **mpeg** ）来按照内容类型显示不同的邮件部分（例如 **application/postscript** 、 **image** 、 **audio** 和 **video** ）。

别名

elm 有其自己的别名系统，该系统支持个人别名和系统级别名。个人别名针对于单个用户；系统别名可用于系统别名所在系统上的所有用户（请参阅 *newalias*(1) ）。通过执行“消息菜单” **a** （别名）命令，可访问“别名菜

单”。可以创建并保存当前消息的别名，创建并检查其他别名，以及将消息发送到一个或多个别名。

别名长度限于 2500 字节。如果要创建长于 2500 字节的组别名，请要求系统管理员在 **sendmail** 系统别名文件 **/etc/mail/aliases** 中为您创建（请参阅 *sendmail*(1M)）。

交互模式菜单和命令

本节将首先介绍“消息菜单”，它是交互模式的主屏幕。其他菜单将按照字母顺序进行介绍。

消息菜单

消息索引显示在“消息菜单”上。可以使用以下命令来处理和发送消息。某些命令使用一系列提示来完成其操作。可以使用 **Ctrl-D** 组合键来取消其操作。

其命令包括：

!*command* Shell 转义符。将 *command* 发送到由 **shell** 字符串变量定义的 Shell，而不退出 **elm** 。

显示有关当前消息的所有已知信息。

$ 将消息重新同步，而不退出 **elm** 。如果任何消息带有删除标记，系统将询问您是否要将其删除。如果已删除任何消息或者任何状态标志已更改，消息将写回到邮箱文件中。所有标记将被删除。

% 显示当前消息的返回地址计算值。

***** 将当前消息指针设置为指向最后一条消息。

+ 显示下一个消息索引页（如果可行）。

- 显示上一个消息索引页（如果可行）。

/*pattern* 模式匹配。在当前消息索引的 *from* 和 *subject* 字段中搜索 *pattern* 。搜索在当前消息处开始，在索引的开头自动分行。当前消息指针设置为指向第一条匹配的消息。大写和小写将同等对待。

//*pattern* 模式匹配。在当前文件夹的所有行中搜索 *pattern* 。搜索在当前消息处开始，在文件夹的开头自动分行。当前消息指针设置为指向第一条匹配的消息。大写和小写将同等对待。

< 日历。在消息中扫描日历条目，并将其添加到日历文件中。日历条目定义为前几个非空白字符为 **->** 的行，如下所示：

->*calendar-entry*

在将该条目添加到日历文件之前，将删除分隔符 **->** 和两边的空格。得到的空白行将被忽略。可以在 **elmrc** 文件中或使用“选项菜单”定义日志文件名。

= 将当前消息指针设置为指向第一条消息。

> 保存在文件夹中。与“消息菜单” **s** （保存）命令相同。

?*key* ... 有关键的帮助。显示每个 *key* 的单行功能说明。 **?** 显示每个可用命令的摘要。句点 **(.)** 将您返回到“消息菜单”。

@ 显示在当前屏幕上索引的消息的摘要。

| 根据需要通过其他过滤器对当前消息或一组带标记的消息进行管道输出。使用由 **shell** 字符串变量定义的 Shell。

n 新的当前消息。将当前消息指针更改为指向索引为 *n* 的消息。如果该消息不在标头的当前页上，则将显示适当的页。

Return 阅读当前的消息。系统将清除屏幕，由 **pager** 字符串变量定义的寻呼将显示当前消息。

a 别名。切换到“别名菜单”。

b 弹转邮件。它类似于转发邮件，不同之处在于您不编辑消息，且返回地址设置为初始发件人的地址，而不是您的地址。

c 更改文件夹。该命令用于更改其消息显示在“消息菜单”上的文件。系统将要求您提供文件名。文件必须是消息格式；否则 **elm** 会异常中止。可以将惯用的通配符以及以下特殊名称用于您的 Shell：

! 收件箱文件夹。

> 已接收邮件文件夹，由 **receivedmail** 字符串变量定义。

< 已发送邮件文件夹，由 **sentmail** 字符串变量定义。

. 先前使用的文件夹。

@*alias* 与 *alias* 别名关联的登录名的缺省文件夹。

=*filename* 由 **maildir** 字符串变量定义的目录中的文件。

C 复制消息。将当前消息或一组带标记的消息另存到一个文件夹。系统将提示您指定文件名，并将提供缺省值。缺省值是 **maildir** 目录中带有所发送的消息集中第一条消息的发件人的用户名的文件。所有标记均会被清除。与 **>** 和 **s** 命令不同，将不会给消息添加删除标记，并不移动当前消息指针。

d 删除。给当前消息添加删除标记。另请参阅 **Ctrl-D** 、 **u** 和 **Ctrl-U** 。

Ctrl-D 删除。给 发件人**:** 和 主题**:** 标头条目中包含指定模式的消息添加删除标记。另请参阅 **d** 、 **u** 和 **Ctrl-U** 。

e 编辑。用于通过由 **editor** 字符串变量定义的编辑器实际编辑当前邮件文件夹。退出编辑器时， **elm** 会将邮件文件夹重新同步（请参阅 **$** 命令）。

f 转发当前消息。系统将询问您是否要编辑外发的消息。如果回答 **y** ，由 **prefix** 字符串变量定义的字符将作为前缀添加到消息的每一行，并且将调用由 **editor** 字符串变量定义的编辑器来编辑消息。如果回答 **n** ，则将不添加字符前缀，并且不调用编辑器。在任一情

况下，系统将提示您指定 收件人**:** 收件人，并允许您编辑 主题**:** 标头条目，而且在 **askcc** 布尔变量是 **ON** 的情况下，将提示您指定 抄送**:** 收件人。

如果 **userlevel** 数值变量为 **1** （中级）或 **2** （专家级），并且该会话中有先前已发送或忘记的消息，系统将询问是否要

撤回上次保留的邮件**? (y/n)**

如果回答 **y** ，先前的消息将返回到发送缓冲区。如果回答 **n** ，则会将当前消息复制到发送缓冲区，并附加您的签名文件（如果有）。

如果您在上面选择编辑外发消息，此时将调用编辑器。退出编辑器或者不调用编辑器时，将显示“发送菜单”。

g 群组回复。回复将通过 收件人**:** 自动发送给消息的发件人，并通过 抄送**:** 发送给所有初始 收件人**:** 和 抄送**:** 收件人。否则，该操作与 **r** 命令的操作相同。

h 与 **Return** 相同，不同的是消息将随所有标头一起显示。

j 下移。将当前消息指针下移到下一条消息。

J 下移。将当前消息指针下移到下一条未删除的消息。

k 上移。将当前消息指针上移到上一条消息。

K 上移。将当前消息指针上移到上一条未删除的消息。

l (ell) 将显示的消息限制为包含特定字符串值的消息。系统将提示您 输入选择条件**:** 。要设置、添加或清除限制条件，请键入以下命令之一:

all 清除所有条件并恢复常规的显示。

from *string* 限制为 发件人**:** 标头中包含 *string* 的条目。

subject *string* 限制为 主题**:** 标头中包含 *string* 的条目。

to *string* 限制为 收件人**:** 标头中包含 *string* 的条目。

可以通过重复 **l** 命令来添加条件。

Ctrl-L 刷新屏幕。

m 邮件。将邮件发送到一个或多个地址。系统将提示您指定 收件人**:** 收件人、 主题**:** 以及（当 **askcc** 布尔变量是 **ON** 时） 抄送**:** 收件人。

如果 **userlevel** 数值变量为 **1** （中级）或 **2** （专家级），并且该会话中有先前已发送或忘记的消息，系统将询问是否要

撤回上次保留的邮件**? (y/n)**

如果回答 **y** ，先前的消息将返回到发送缓冲区。如果回答 **n** ，则将签名文件（如果有）复制到发送缓冲区中。

然后，将调用由 **editor** 字符串变量定义的编辑器。退出编辑器后，将显示“发送菜单”。

n 下一条消息。将当前消息指针指向下一条消息，并显示与 **Return** 命令相同的消息。

o 选项。调用“选项菜单”，可用于更改某些配置选项。可更改的选项由 **configoptions** 字符串变量定义。

p 打印。使用由 **print** 字符串变量定义的命令打印当前消息或一组带标记的消息。当前消息指针不会移动。带标记的消息仍然带有标记。

q 退出。正常终止，并按照定义的个人首选项执行消息清理。可以选择实际删除带有删除标记的消息。对于收件箱，可以选择将未删除的邮件保留在邮箱中，或者将其移到由 **receivedmail** 字符串变量定义的已接收邮件文件夹中。

如果 **ask** 布尔变量是 **ON** ，系统可能会向您询问以下问题。所述的操作均将在您回答所有相关问题后执行。

是否删除邮件**? (y/n)**

如果消息带有删除标记，则将询问该问题。缺省答案由 **alwaysdelete** 布尔变量提供（**N** 表示 **y** （是）， **OFF** 表示 **n** （否））。

如果回答 **y** ，则将删除所有带删除标记的消息。

如果回答 **n** ，则会将所有带删除标记的消息恢复到其先前的“已读”、“未读”或“新到”状态。

是否将已读邮件移至 **received"** 文件夹中**? (y/n)"**

如果您正在阅读收件箱，或者邮件已读，则将询问该问题。缺省答案由 **alwaysstore** 布尔变量提供（ ON 表示 **y** （是）， **OFF** 表示 **n** （否））。

如果回答 **y** ，未删除的已读消息将移到由 **receivedmail** 字符串变量定义的文件，并且还将询问下一个问题。

如果回答 **n** ，所有未删除的消息将返回到收件箱，并且将不询问下一个问题。

是否将未读邮件保留在收件邮箱中**? (y/n)**

如果您正在阅读收件箱，对 是否将已读邮件移至... 问题回答 **y**

（或者未询问该问题），并且有未读的消息，则将询问该问题。缺省答案由 **alwayskeep** 布尔变量提供（ ON 表示 **y** （是）， **OFF** 表示 **n** （否））。

如果回答 **y** ，所有未删除的未读消息（无论新旧）均将返回到收件箱。

如果回答 **n** ，所有未删除的未读消息均将移到由 **receivedmail** 字符串变量定义的文件夹。

如果 **ask** 布尔变量是 **OFF** ，则将分别采用 **alwaysdelete** 、 **alwaysstore** 和 **alwayskeep** 布尔变量的值作为问题的答案（不显示）。

Q 快速退出。它与在 **ask** 布尔变量设置为 **OFF** 时执行 **q** 命令等效。

r 回复当前消息的发件人。如果 **autocopy** 布尔变量是 **OFF** ，系统将询问您是否应该将源消息复制到编辑缓冲区中。如果该变量是 **ON** ，则将自动复制消息。如果已复制，消息中所有行的前面均将添加由 **prefix** 字符串变量定义的前缀字符串。 收件人**:** 标头设置为消息的发件人（或者 答复至**:** 标头中的地址，如果已设置）， 主题**:** 设置为消息的主题，前面加有 **Re:** 前缀，并供您进行编辑。如果 **askcc** 布尔变量是 **ON** ，则将提示您指定 抄送**:** 收件人。然后，将调用由 **editor** 字符串变量定义的编辑器。退出编辑器后，将显示“发送菜单”。

s 保存在文件夹中（与 **>** 相同）。将当前消息或一组带标记的消息另存到一个文件夹。系统将提示您指定文件名，并将提供缺省值。缺省值是 **maildir** 目录中带有所发送的消息集中第一条消息的发件人的登录名的文件。将清除所有标记并给消息添加删除标记。当前消息指针将移到上一个已保存消息后的第一个未删除消息。

t 标记切换。为稍后的操作标记当前消息，并将当前消息指针移到指向下一条未删除的消息。该操作可以是 **|** 、 **C** 、 **p** 和 **s** 之一。

或者，删除带标记消息的标记。另请参阅 **Ctrl-T** 命令。

T 标记切换。为稍后的操作标记当前消息，并保留在当前的消息处。该操作可以是 **|** 、 **C** 、 **p** 和 **s** 之一。

或者，删除带标记消息的标记。另请参阅 **Ctrl-T** 命令。

Ctrl-T 标记包含指定模式的所有消息。或者，删除所有带标记消息的标记。

如果任何消息当前带有标记，则将询问您是否应该删除标记。回答 **y** 将删除旧标记；回答 **n** 将保留标记。在任一情况下，系统都将提示您在每条消息的 发件人**:** 或 主题**:** 行中指定要匹配的字符串。将标记所有匹配条件的消息。如果输入空字符串（单独输入回车），则将不再标记其他消息。

u 取消删除。删除当前消息的删除标记。另请参阅 **d** 、 **Ctrl-D** 和 **Ctrl-U** 。

Ctrl-U 取消删除。删除在 发件人**:** 和 主题**:** 标头条目中包含指定模式的所有消息的任何删除标记。另请参阅 **d** 、 **Ctrl-D** 和 **u** 。

v 查看附件。对当前消息调用“附件查看菜单”。

x 退出。退出而不更改邮箱。如果更改（如删除）未决，则将询问您是否可以放弃更改。如果回答 **y** ，系统将放弃更改，而程序会终止。如果回答 **n** ，则将放弃退出，而返回到“消息菜单”命令提示符处。

X 立即退出而不更改邮箱。放弃所有未决的更改。

消息索引

当前文件夹中的消息在“消息菜单”中进行索引，每行显示一个索引，格式如下：

sssnum mmm d from **(***lines***)** *subject*

其定义如下：

sss 包含三个字符的状态字段，在“消息状态”小节中介绍。

num 消息索引序号。

mmm 最后一个 日期**:** 标头条目或 发件人消息标头中的月份。

d 最后一个 日期**:** 标头条目或 发件人消息标头中的日期。

from 最后一个 发件人**:** 标头条目或 发件人消息标头中的发件人名称。

lines 消息的行数。

subject 第一个 主题**:** 标头条目的主题说明，为适合屏幕显示而被截断。

当前消息索引条目在逆向视频中突出显示，或者在左边缘处标有箭头 (**->**) 。请参阅“选项”小节中的 **-a** 选项以及 ELM 配置部分的 **arrow** 字符串变量。

消息状态

每个消息索引条目的前三个字符描述消息的状态。每个字符可以是空白或者下述值之一（按优先权降序排列）。

当消息包含特定类型集的多个状态标志时，索引行上将显示具有最高优先权的指示符。例如，如果表单消息 (**F**) 也标记为公司机密 (**C**) ，则将显示 **C** 而不是 **F** 状态字符。

第一列：可变状态

D 已删除。该消息带有删除标记。

E 已到期。 过期**:** 标头条目中指定的日期早于当前日期。 **elm** 接受以下日期格式：

Mon, 11 Jun 90 （由 **elm** 在“标头菜单”中生成的格式）
Jun 11, 90
11 Jun, 90
9006111324GMT （ISO X.400 格式： *YYMMDDhhmm* ）

N 新消息。该消息在上次 **elm** 会话之后或在当前会话中收到。尚未阅读该消息。

O 旧消息。该消息在上次 **elm** 会话之前或之中收到。它在上一次会话中标记为 **N** ，并且未被阅读。

空白。该消息已读。

第二列：永久状态

C 机密。存在 **Sensitivity: 3** 标头条目。该消息按照 ISO X.400 标准的规定被视为公司机密。可以使用“标头菜单”的用户定义选项为外发邮件指定该值。

U 紧急。该消息包含 优先级**:** 标头条目。

P 私密。存在 **Sensitivity: 2"** 标头条目。该消息按照 ISO X.400 标准的规定被视为私密。可以使用“标头菜单”的用户定义选项为外发邮件指定该值。

A 操作。该消息包含 操作**:** 标头条目。

F 表单。该消息是 **elm** 表单消息。该消息包含 内容**-**类型**: mailform** 标头条目。

M MIME。该消息或其附件使用的是可以使用 **metamail** 显示的 MIME 格式。

? MIME。该消息或其附件使用的是版本不受支持的 MIME 格式。

空白。常规状态。

第三列：带标记的状态

+ 带标记。某些命令将带标记的消息作为一个组进行处理。请参阅“消息菜单”小节中的 **t** 和其他命令。

空白。该消息不带标记。

内置编辑器

使用 **builtin** 内置编辑器编写外发消息时，将以一个空行提示您输入文本行。输入句点 (**.**) 结束消息，然后继续使用“发送菜单”进行操作。

内置编辑器命令是以转义符（由 **escape** 字符串变量定义）开头的行。缺省的转义符是波浪符 (˜) 。

注意：当波浪符是行中的第一个字符时，某些远程登录程序将该波浪符用作其缺省转义符。（由于波浪符不输出，您可以将其识别）。通常，当您输入另一个程序无法识别的字符或者输入另一个波浪符时，将传输波浪符。有关详细信息，请参阅程序文档。

内置编辑器命令包括：

˜! [*command*] 执行 Shell *command* （如果已给定，如 **˜!ls** 所示），或者使用由 **shell** 字符串变量定义的 Shell 启动一个互动 Shell。

˜< *command* 执行 Shell *command* 并将该命令的输出放入编辑器缓冲区。例如，“**˜< who**”在消息中插入 **who** 命令的输出。

˜? 输出简短的帮助菜单。

˜˜ 使用单个波浪符 (˜) 开始一行。

˜b 提示对密件抄送 (密件抄送**:**) 列表的更改。

~c 提示对抄送 (抄送**:**) 列表的更改。

~e 对消息调用为 **easyeditor** 字符串变量定义的编辑器（如果可能）。

~f [*options*] 添加指定消息列表或当前消息。它使用 **readmail** ，这意味着所有 **readmail** 选项均可用（请参阅 *readmail*(1) ）。

~h 提示对所有可用标头 （收件人**:** 、 抄送**:** 、 密件抄送**:** 和 主题**:** ）的更改。

~m [*options*] 与 **~f** 相同，但是每行带有当前前缀。请参阅 **prefix** 字符串变量。

~o 提示指定要用于消息的编辑器的名称。

~p 打印到目前为止键入的消息。

~r *filename* 包括（读入）指定文件的内容。

~s 提示对 主题**:** 行的更改。

~t 提示对 收件人**:** 列表的更改。

~v 对消息调用为 **visualeditor** 字符串变量定义的编辑器（如果可能）。

别名菜单

“别名菜单”使用“消息菜单” **a** 命令来调用。别名文件的源文本存储在 **$HOME/.elm/aliases.text** 文件中。可以直接或使用以下命令编辑该文件。

别名当前编译到您的数据库中，系统数据库显示在与“消息菜单”类似的索引列表中。条目格式在“别名索引”小节中进行介绍。索引按照由 **aliassortby** 字符串变量定义的顺序进行排序。

其命令包括：

$ 通过运行 **newalias** 从文本文件重建数据库来将别名文本文件和别名数据库重新同步。将删除带有删除标记的别名，删除带标记别名的标记，并且识别新的和已更改的别名。别名屏幕将更新，以反映这些更改。

+ 在适当时显示下一个别名索引页。

- 在适当时显示上一个别名索引页。

/*pattern* 模式匹配。在别名列表的别名和用户名字段中搜索 *pattern* 。搜索在当前别名处开始，在别名列表的开头自动分行。当前别名指针设置为指向第一个匹配的别名。大写和小写将同等对待。

//*pattern* 模式匹配。在别名列表的所有字段（别名、用户名、注释和地址）中搜索 *pattern* 。搜索在当前别名处开始，在别名列表的开头自动分行。当前别名指针设置为指向第一个匹配的别名。大写和小写将同等对待。 **/***pattern* 模式匹配。该命令可用于在别名列表的所有别名和用户名条目中进行搜索（从当前别名开始，一直到末尾）。如果模式的第一个字符是 **/** ，则搜索中还将包括注释和完全展开的地址字段。搜索不区分大小写。这样，可以在存在大量别名的情况下查找特定的别名。

?*key* ... 有关键的帮助。显示每个 *key* 的单行功能说明。 **?** 显示每个可用命令的摘要。句点 (**.**) 将您返回到“别名菜单”。

a 别名当前消息。它可用于创建以当前消息的返回地址作为其地址字段的的别名。它将提示您指定唯一的别名名称，并允许编辑注释和地址字段。

c 更改当前用户别名。别名字段的旧值用作新值提示中的缺省值。不能更改别名名称。如果别名名称是多别名记录的别名名称，则会将其从该记录中删除，并将其作为单独的记录存储。旧的别名标记为 **N** 。更改将在下次别名重新同步后生效。

d 给当前用户别名添加删除标记。使用 **q** 、 **r** 或 **i** 命令退出“别名菜单”或者使用 **$** 命令将别名数据库重新同步时，将执行删除。（不能以这种方式删除系统别名）。

Ctrl-D 删除符合指定搜索模式的用户别名。

e 使用在 **editor** 字符串变量中定义的编辑器编辑 **aliases.text** 文件。完成编辑时，将重新同步别名（请参阅 **$** 命令）。

f 显示完全展开的别名。将完全展开并显示当前选定的别名。

i,**I** 请参阅“别名菜单” **q** 和 **Q** 命令。

j 下移。将当前别名指针下移到下一个别名。

J 下移。将当前别名指针下移到下一个未删除的别名。

k 上移。将当前别名指针上移到上一个别名。

K 上移。将当前别名指针上移到上一个未删除的别名。

l (ell) 将显示的别名限制为特定类型或包含特定字符串值的别名。系统将提示您 输入选择条件**:** 。要设置、添加或清除限制条件，请键入以下命令之一:

- **all** 清除所有条件并恢复常规的显示。
- **alias** *string* 限制为包含 *string* 的别名名称。
- **name** *string* 限制为包含 *string* 的完整名称（名字和姓氏）。
- **group** 限制为组别名（可以包含系统和用户别名）。
- **person** 限制为个人别名（可以包含系统和用户别名）。
- **system** 限制为系统别名（可以包含组和个人别名）。
- **user** 限制为系统别名（可以包含组和个人别名）。

可以通过重复 **l** 命令来添加条件。

Ctrl-L 刷新屏幕。

m 通过邮件发送到当前别名或者一组带标记的别名。相应的展开地址放置在 收件人**:** 标头条目中，并且会继续按照与“消息菜单” **m** （邮件）命令相同的方式进行。将清除标记。

n 创建用户别名。 **elm** 将先提示指定唯一的别名名称，再提示指定地址。所提供的信息将添加到您个人的 alias_text 文件 (**$HOME/.elm/aliases.text**) ，然后再添加到数据库中。

q 退出。返回到“消息菜单”。如果别名带有删除标记，系统将询问您是否要将其删除。将保留别名索引指针。如果更改了别名文本文件，数据库将重新同步。

Q 退出。返回到“消息菜单”。如果别名带有删除标记，则将保留该标记，并且不删除别名。将保留别名索引指针。如果更改了别名文本文件，数据库将重新同步。

r,**R** 请参阅“别名菜单” **q** 和 **Q** 命令。

t 为稍后的操作标记当前别名，并将当前别名指针移到指向下一个未删除的别名。该操作可以是 **c** 、 **m** 或 **n** 之一。

或者，删除带标记别名的标记。另请参阅 **Ctrl-T** 命令。

T 标记。为稍后的操作标记当前别名，并保留在当前的别名处。该操作可以是 **c** 、 **m** 或 **n** 之一。

或者，删除带标记别名的标记。另请参阅 **Ctrl-T** 命令。

Ctrl-T 为稍后的操作标记包含指定模式的所有别名。该操作可以是 **c** 、 **m** 或 **n** 之一。或者，删除所有带标记别名的标记。

如果任何别名当前带有标记，则将询问您是否应该删除标记。回答 **y** 将删除旧标记；回答 **n** 将保留标记。在任一情况下，系统将提示您在每个别名的别名名称或用户名字段中输入要匹配的字符串。将标记所有匹配条件的别名。如果输入空字符串（单独输入回车），则将不再标记其他别名。

u 取消删除。删除当前别名的删除标记。另请参阅 **d** 、 **Ctrl-D** 和 **Ctrl-U** 。

Ctrl-U 取消删除。删除在 发件人**:** 和 主题**:** 标头条目中包含指定模式的所有消息的任何删除标记。另请参阅 **d** 、 **Ctrl-D** 和 **u** 。

v 查看。显示当前别名的 *address-list* 。

x 退出别名菜单，而不处理任何删除。带有删除标记的别名将取消标记，并且即使添加了别名， **newalias** 仍不会运行。

别名索引

当前数据库中的别名在“别名菜单”中进行索引，每行显示一个索引。数据库值在 *newalias*(1) 中定义。

ssnum fullname[**,** *comment*] *type* [**(S)**] *alias*

其定义如下:

ss 包含两个字符的状态字段。第一个字符可能是:

D 删除。该别名带有删除标记。

N 新别名。该别名是新的或在别名文本文件中进行了更改，但未包括在当前数据库中。需要进行重新同步。

空白。该别名包含在当前数据库中。

第二个字符可能是:

+ 标记。该别名带有标记。

空白。该别名不带有标记。

num 该别名的索引号。

fullname 别名的完整名称，即在展开的数据库中使用的形式。其格式如下:

firstname lastname

firstname 名字，来自别名数据库。

lastname 姓氏，来自别名数据库。

comment 注释，来自别名数据库。

type 别名的类型。对于包含单个地址的别名为 个人；对于包含两个或更多个地址的别名为 组。

(S) 如果存在，该条目来自 **elm** 系统别名数据库。如果不存在，该条目来自您的个人别名数据库。

alias 别名名称。如果记录包含多个别名名称，每个名称都会有一个索引条目。

附件配置菜单

“附件配置菜单”使用“附件发送菜单” **a** （添加）或 **m** （修改）命令来调用。该菜单显示附件的缺省或当前配置。如果使用 **a** 命令调用，它将自动提示输入文件名。其命令包括:

d 说明。其值放入 内容**-**说明**:** 正文部分标头条目。缺省值是文件名。

e 内容传输编码。这是一种文件编码方法，它允许通过各种邮件传输代理传递文件。可用选项包括:

7bit 未编码的常规 **US-ASCII** 文本。这是缺省的编码参数。

8bit 未编码的 8 位字符，包含高次位集。

quoted-printable
包含控制字符和高次位字符的文本，转换为 **=***hh* 形式的字符串，其中 *hh* 是字符的十六进制表示形式。行尾的 **=** 指示源行已分开成两行。

base64 其位以 6 位组进行编码且按数值顺序呈现为 **US-ASCII** 字符 **A–Z** 、 **a–z** 、 **0–9** 、 **+** 和 **/** 的任何文件类型。最后一行可以通过 **=** 字符填充到 4 个字符的倍数。

binary 未编码的二进制数据。

其值放入 内容**-**传输**-**编码**:** 正文部分标头条目。缺省值是 **7bit** 。

f 文件名。要附加的文件的名称。 **elm** 检查该文件并相应地设置内容传输编码、内容处理和内容类型的值。

p 内容处理。其值放入 内容**-**部署**:** 正文部分标头条目。缺省值是 **attachment; filename=***filename* 。

t 内容类型。文件的类型以及支持参数，其格式如下:

*type***/***subtype* [**;** *parameter*]...

type 可以是 RFC 1521 所定义的 **application** 、 **audio** 、 **image** 、 **message** 、 **text** 或 **video** 之一。虽然 **multipart** 也是有效的类型，但不能直接指定该类型； **elm** 根据需要提供该类型，并处理包含它的消息。其值放入 内容**-**类型**:** 正文部分标头条目。缺省值是:

text/plain; charset=US-ASCII

下面将介绍一些常用的条目。有关其他信息，请参阅 METAMAIL 配置部分。

text/*subtype* [**; charset=***charset*]
它是较易于阅读的文本，对于可能的子类型 **richtext** 或 **html** ，可以使用嵌入的文本字符进行格式设置。缺省的 *subtype* 是 **plain** （未进行任何形式的格式设置）。缺省的 *charset* 是 **US-ASCII** 。

application/octet-stream
这对于程序二进制文件等文件或者包括控制字符或高次位集字符的文件是一种非常全面的格式。

application/postscript
文件可以使用配备 PostScript 的打印机或查看器显示。

message/rfc822
它指定文件使用消息格式，如“术语”小节所述。

image/jpeg, **image/gif**
它们是需要使用显示程序的图片格式。

audio/basic
这是需要使用再现程序的音频格式。

video/mpeg
这是需要使用再现程序的音频/视频格式。

附件发送菜单

“附件发送菜单”使用“发送菜单” **a** 命令来调用。该菜单显示将在消息中发送的附件的列表，每行显示一个附件，如“附件索引”小节所述。

其命令包括：

a 添加附件。调用“附件配置菜单”并提示指定文件名。

d 删除附件。

e 编辑附件。调用与附件关联的编辑器（如果该附件可编辑）。

j 将当前附件指针下移到下一个附件。

k 将当前附件指针上移到上一个附件。

Ctrl-L 刷新屏幕。

m 修改附件的属性。调用“附件配置菜单”。

p 打印附件。请参阅“消息菜单” **p** （打印）命令。

q 退出。返回到“发送菜单”。

s 保存附件。请参阅“消息菜单” **C** （复制）命令。

附件查看菜单

“附件查看菜单”使用“发送菜单” **v** 命令来调用。该菜单显示文件夹消息中附件的列表，每行显示一个附件，如“附件索引”小节所述。

其命令包括：

Return 显示当前附件。

j 将当前附件指针下移到下一个附件。

k 将当前附件指针上移到上一个附件。

Ctrl-L 刷新屏幕。

p 打印附件。请参阅“消息菜单” **p** （打印）命令。

q 退出。返回到上一个附件级别或“消息菜单”。

s 保存附件。附件以接收时的形式保存，这与“消息菜单” **s** （保存）命令相同。

v 查看子附件列表（如果有）。

附件索引

附件在“附件发送菜单”和“附件查看菜单”上按以下格式列出：

num filename (*size*) *format* [*encoding*]

其定义如下:

num	该附件的索引号。
filename	附加文件的名称。
size	附件的字节数大小，从文件或消息中进行计算。
type/subtype	附件的类型和子类型。该值位于 内容**-**类型**:** 标头。
encoding	编码类型。该值位于 内容**-**传输**-**编码**:** 标头。

文件附件

elm 仍然支持包括以下格式的一个或多个关键字行的旧方法:

[include *file contenttype/subtype* [*encoding*]**]**

需要随邮件附加的文件包括在消息正文中，并且成为消息的一部分。位于 **include** 行之前、之间和之后的文本行包括在不同的附件中。

编码参数是可选的。有关有效 *contenttype/subtype* 和 *encoding* 参数值的信息，请参阅 RFC 1341。

例如：要将文件 **cartoon.gif** 包括在消息中，并使用 base-64 编码，请使用以下行:

[include cartoon.gif image/gif base64]

或者，如果要包括文本文件 **foo.txt** （它包含纯 ASCII），请使用以下行:

[include foo.txt text/plain]

消息加密

为了增强安全性和私密性，可以使用以下关键字行将消息加密： **[encode]** 和 **[clear]** 。

考虑以下外发消息:

Hi Tom,
How are you doing?
[encode]
 This is a private message!!
[clear]
 Keep in touch.
  - Jerry

[clear] 行表示从 **[encode]** 到 **[clear]** 的文本块的加密的结尾。

以上消息在键入编辑器时是可读的，并且一旦确认需要发送消息， **elm** 邮件程序将提示您:

输入加密密钥： @

它接收密钥（8 个更少个字符），而不回显到屏幕上。您将需要重新输入该密钥。如果已经启用复制选项，则程

序将以加密形式保存消息的副本。这有助于确保邮件归档的私密性和安全性。

如果邮件程序没有提示您输入加密密钥，则不会自动输入 **[encode]** 作为该行前 8 个字符。同样，还应输入 **[clear]** 作为该行的前 7 个字符，指示加密的结尾。

在另一端，系统将用以下消息提示该邮件的收件人（应该是 **elm** 程序的用户）输入解密密钥：

输入解密密钥： **@**

如果解密密钥正确，该程序会将邮件解密，并以可读的形式显示每一行。如果解密密钥不正确，则将显示加密形式的相同消息。

注意：当前解密不支持对加密的邮件进行 **pipe** 或 **print** 处理。

邮件归档

该功能可用于指定需要归档的消息内容（假定已启用复制）。

要指定要归档的消息的最后一行，需要在消息正文中输入 **[nosave]** 或 **[no save]** 关键字行。

保存的邮件将不包含 **[nosave]** 或 **[no save]** 关键字行之后的消息。但是，外发的邮件将包含除关键字行之外的所有消息。

标头菜单

“标头菜单”使用“发送菜单” **h** 命令来调用。它用于在消息中添加、更改和删除一组常用的标头条目。通常，如果项目为空，则不会包含在消息中。

其命令包括：

Return	返回到“发送菜单”。
!*command*	Shell。使用由 **shell** 字符串变量定义的 Shell 执行 *command* 。
a	操作**:** 标头。输入任意字符串。如果该条目存在于收到的消息中， **elm** 将在消息索引的“永久状态”列中显示 **A** 。
b	密件抄送**:** 标头。输入别名和实际地址的列表。别名将展开，并显示为地址和用户名。
c	抄送**:** 标头。输入别名和实际地址的列表。别名将展开，并显示为地址和用户名。
d	域化。将非 Internet 地址转换为 Internet 格式。 UUCP **!** 格式 (*host***.***domain***!***user*) 成为 Internet @ 格式 (*user*@*host***.***domain*)。如果省略了 **.***domain* ，则缺省为 **.uucp** 。
e	过期**:** 标头。输入 1 到 56（8 周）之间的任何数字值。如果该条目存在于收到的消息中，当经过计算的日期时， **elm** 将在消息索引的“可变状态”列中显示 **E** 。
i	答复至**:** 标头。输入字符串。

n 优先级**:** 标头。输入优先权名称。如果 **precedences** 字符串变量已设置且不为空，则名称必须是由该变量定义的名称。如果该名称与优先级相关联，并且 优先级**:** 标头为空，则优先级值将插入 优先级**:** 标头中。如果 **precedences** 为空或者未设置，则可以输入任意值。

如果优先权名称匹配 **sendmail** 配置文件 **/etc/mail/sendmail.cf** 中定义的名称，则将相应地修改传输优先级。如果没有匹配项，则将不更改优先级。

p 优先级**:** 标头。输入字符串。如果该条目存在于收到的消息中， **elm** 将在消息索引的“永久状态”列中显示 **U**

r 答复至**:** 标头。输入个人别名或单个地址。如果已存在， **elm** 和其他邮件程序在选择回复地址（“消息菜单” **r** （回复）命令）时将使用该标头，而不是 发件人**:** 标头。

s 主题**:** 标头。输入字符串。

t 收件人**:** 标头。输入别名和实际地址的列表。别名将展开，并显示为地址和用户名。

u 用户定义的标头。按照以下格式定义您自己的标头条目:

*header-name***:** *header-string*

*header-name***:** 不得包含空白字符。可以使用该命令创建 **Sensitivity:** 标头条目（如“消息状态”小节所述），或者创建不同的标头（仅限于一个）。有关通过其他方法包括用户定义标头条目的信息，请参阅 HEADER FILE 部分。

选项菜单

“选项菜单”使用“消息菜单” **o** 命令来调用。它显示由 **configoptions** 字符串变量定义的选项，您可以在 **elm** 运行过程中修改这些选项。输入适当的字母（大写或小写），其后接右括号 ())，然后按照屏幕上的说明进行操作。下面列出了全部的选项提示以及相应的变量。缺省标记标有 ***** 。

A)箭头光标	**arrow** 字符串变量。 *****
B)副本边界	**prefix** 字符串变量。
C)日历文件	**calendar** 字符串变量。 *****
D)显示邮件时使用	**pager** 字符串变量。 *****
E)编辑器	**editor** 字符串变量。 *****
F)文件夹目录	**maildir** 字符串变量。 *****
H)保留已发送邮件	**copy** 布尔变量。
J)答复编辑器	**alteditor** 字符串变量。
K)在分页程序之后暂停	**promptafter** 布尔变量。
别名排序	**aliassortby** 字符串变量。
M)菜单显示	**menu** 布尔变量。 *****
N)只显示人名	**names** 布尔变量。 *****

O)外发邮件存入	**sentmail** 字符串变量。 *
P)打印邮件使用	**print** 字符串变量。 *
R)答复复制邮件	**autocopy** 布尔变量。
S)排序条件	**sortby** 字符串变量。 *
T)文本编辑器 **(˜e)**	**easyeditor** 字符串变量。
U)用户级别	**userlevel** 数值变量。 *
V)可视化编辑器 **(˜v)**	**visualeditor** 字符串变量。 *
W)想要抄送**:** 提示	**askcc** 布尔变量。
Y)您的全名	**fullname** 字符串变量。 *
Z)签名破折号	**sigdashes** 布尔变量。

注意：该菜单显示定义集中的前 *screen-height***-6** 个行。 *screen-height* 是屏幕上的文本行行数。如果未显示选项，则无法对其进行修改。

完成后，输入以下值之一：

> 将当前配置值保存在配置文件 **$HOME/.elm/elmrc** 中。如果不存在该文件，则将其创建。这是创建配置文件的简便方法，您可以直接编辑，也可以使用“选项菜单”进行编辑。

i,**I** 返回到“消息菜单”。

q,**Q** 返回到“消息菜单”。

x,**X** 直接退出 **elm** 而不更改邮箱。放弃所有未决的更改。

发送菜单

在执行“消息菜单” **f** 、 **g** 、 **m** 或 **r** 命令或者“别名菜单” **m** 命令之后，外发消息准备通过邮件发送时，将调用“发送菜单”。

其命令包括：

!*command* Shell。执行 Shell 命令。请参阅“消息菜单” **!** (shell) 命令。

a 附件。调用“附件发送菜单”。

c 复制。复制到文件。请参阅“消息菜单” **C** （复制）命令。

e 编辑。调用由 **alteditor** 字符串变量定义的编辑器来修改消息。

f 忘记。不用发送消息。在用户级别 **1** 和 **2** ，当执行后继的“消息菜单” **f** 、 **g** 、 **m** 或 **r** 命令或者“别名菜单” **m** 命令时，消息可能会返回到发送缓冲区。

h 编辑标头条目。调用“标头菜单”。

m 制作表单。将消息转换为表单消息格式。请参阅“表单消息”部分。只有当 **forms** 布尔变量为 **ON** 且 **userlevel** 数值变量为 **1** 或 **2** 时，该命令才可用。

s 发送。发送消息。

表单消息

elm 的这一特有功能可用于撰写和回复表单消息。

创建表单消息

- 在 **elmrc** 文件中，设置 **forms=ON** 。
- 将 **userlevel** 数值变量设置为 **1** （适当经验）或 **2** （专家级）。可以在 **elmrc** 文件或缺省的“选项菜单”中进行该设置。
- 撰写消息时，将使用冒号 **(:)** 后接字段值可用数目的空格或者新行（指示字段可以填充该行的剩余部分）的形式来定义将由收件人填充的字段。

  行上的冒号本身指示将提示收件人进行多行输入。冒号之前不能有空白字符。

  包含冒号的每一行均是一个提示行。在响应过程中，将删除每一行上从最后一个冒号后的第一个非空白字符开始的所有文本，并且为响应字段对该行进行计算。
- 创建消息后，输入“发送菜单” **m** （创建表单）命令来设置特殊格式。然后输入“发送菜单” **s** （发送）命令发送消息。

下面是一个简单表单消息的示例：

On-Line Phone and Address Database
Please fill out and return as soon as possible.
Name:
Manager:
Department: **Division:**
Your home address:
Home phone number:
Thank you for your cooperation.

回复表单消息

接收表单消息时，消息索引条目将带有 **F** 状态字母标记。可以使用 **Return** 或 **h** 命令按照常规方式查看该消息。

要进行回复，请使用“消息菜单” **r** （回复）命令（不能使用“消息菜单” **g** （群组回复）命令）。 **elm** 将提示您填写每个字段，并且在适当情况下显示存在于字段之间的任何文本。以上示例逐行显示；用户输入为斜体：

On-Line Phone and Address Database
Please fill out and return as soon as possible.
Name:*my name*
Manager:*my manager*
Department:*my department*
Division:*my division*
Your home address:*home address*
Home phone number:*phone number*
Thank you for your cooperation.

收到的消息如下所示：

On-Line Phone and Address Database
Please fill out and return as soon as possible.
Name: my name
Manager: my manager
Department: my department **Division: my division**

Your home address: home address
Home phone number: phone number
Thank you for your cooperation.

标头文件

通过 **$HOME/.elm/elmheaders** 文件，可以指定 **X-Organization:** 、 **X-Phone:** 等特殊标头信息。该文件中的非空白行将添加到所有外发邮件中的标头。

elmheaders 文件中的条目应该具有以下格式：

*header-name***:** *header-string*

*header-name***:** 不得含空白字符。 *header-string* 可以在多个行上连续，每个连续行之前加有空白字符，如以下输出所示。

在 **elmheaders** 文件中，可以使用反引号（左撇号）在读取文件时执行 Shell 命令，这样如下格式的条目：

X-Operating-System: ‘uname -a‘

将生成如下标头条目：

X-Operating-System:
HP-UX hpulpc17 B.10.10 A 9000/710 2012505939 two-user license

根据 RFC 822，用户定义的标头名应该以 **X-** 或 **x-** 开头。否则，如果在稍后使用其他某种方法将名称标准化，则存在将其用法覆盖的风险。

定义的标头

RFC 822 和 RFC 1521 中为消息标头定义了以下标头名。

密件抄送:	(822)	抄送:	(822)
注释:	(822)	内容-说明:	(1521)
内容-ID:	(1521)	内容-传输-编码:	(1521)
内容-类型:	(1521)	日期:	(822)
Encrypted:	(822)	发件人:	(822)
答复至:	(822)	Keywords:	(822)
MIME-Version:	(1521)	Message-ID:	(822)
接收:	(822)	References:	(822)
答复至:	(822)	重发-密件抄送:	(822)

重发-抄送:	(822)	重发-日期:	(822)
重发-发件人:	(822)	Resent-Message-ID:	(822)
重发-答复至:	(822)	重发-发件人:	(822)
重发-收件人:	(822)	Return-Path:	(822)
发件人:	(822)	主题:	(822)
收件人:	(822)	X-user-defined:	(822)

其他常用标头:

操作:	Apparently-To:
内容-部署:	内容-长度:
过期:	电子邮件程序:
Newsgroups:	优先级:
优先级:	Sensitivity:
状态:	X-Mailer:

ELM 配置

elm 支持用户通过 **$HOME/.elm/elmrc** 配置文件进行配置。可以使用“选项菜单” **>** 命令来创建该配置文件。它可以包含下述字符串、数值和布尔变量的任意组合。

字符串变量

字符串变量格式如下

string-name **=** *string-value*

下面定义下列字符串变量。

aliassortby 别名索引在“别名菜单”中的排序顺序。可识别的值为:

alias 按别名名称排序。

name 按别名的完整名称排序（姓氏放在前面）。

text 按别名在别名文本文件中的顺序排序。

给值添加 **reverse-** 前缀，以反转排序顺序。缺省值是 **name** 。

alteditor 当 **editor** 字符串变量设置为 **none** 或 **builtin** 时，要用于包括初始文本（回复中的复制消息、任意外发消息中的签名等）的消息的编辑器的名称。如果 **EDITOR** 环境变量已设置且不为空，则缺省值为该变量的值；否则为 **/usr/bin/vi** 。另请参阅 **editor** 字符串变量。

alternatives 从中接收转发邮件的其他计算机和用户名组合的列表。当处理群组回复以确保回复消息不发送到仅会将回复消息转回给您的用户和/或计算机地址时，**elm**

将使用该信息。缺省值为无。

attribution 回复的特性字符串。回复消息并将该消息包含在回复中时，该字符串将放在消息的顶部。字符 **%s** 会替换为初始消息作者的完整名称。缺省值为无。例如：

attribution = %s wrote:

calendar 日历文件的名称。它与“消息菜单” **<** （日历）命令一起使用，在消息中扫描日历条目。缺省值是 **$HOME//calendar** 。

charset 对于 **text/plain** 类型用于 MIME 内容-类型**:** 标头的字符集的名称。它可以是作为 **US-ASCII** 超集的任意 Internet 定义字符集名称。缺省值是 **US-ASCII** 。例如，

Content-Type: text/plain; charset=US-ASCII

compatcharsets 作为 US-ASCII 超集的 Internet 定义字符集列表，可使包含 **charset=US-ASCII** 的消息无需借助于 **metamail** 即可显示。缺省值是包含以下值的字符串：

Extended_UNIX_Code_Packed_Format_for_Japanese
ISO-2022-JP
ISO-8859-1
ISO-8859-2
ISO-8859-3
ISO-8859-4
ISO-8859-5
ISO-8859-7
ISO-8859-8
ISO-8859-9
KOI8-R
Shift_JIS

configoptions 可在“选项菜单”上配置的选项的字符串。按照其显示顺序将每个选项指定为单个字母。缺省值是“**^_cdefsopyv_am_un**”。以下缺省值标有 ***** 。

选项字符包括：

^ 菜单标题。
_ 空白行。
a **arrow** 字符串变量。 *****
b **prefix** 字符串变量。
c **calendar** 字符串变量。 *****
d **pager** 字符串变量。 *****

- **e** **editor** 字符串变量。 *
- **f** **maildir** 字符串变量。 *
- **h** **copy** 布尔变量。
- **j** **alteditor** 字符串变量。
- **k** **promptafter** 布尔变量。
- **l** **aliassortby** 字符串变量。
- **m** **menu** 布尔变量。 *
- **n** **names** 布尔变量。 *
- **o** **sentmail** 字符串变量。 *
- **p** **print** 字符串变量。 *
- **r** **autocopy** 布尔变量。
- **s** **sortby** 字符串变量。 *
- **t** **easyeditor** 字符串变量。
- **u** **userlevel** 数值变量。 *
- **v** **visualeditor** 字符串变量。 *
- **w** **askcc** 布尔变量。
- **y** **fullname** 字符串变量。 *
- **z** **sigdashes** 布尔变量。

displaycharset 显示所支持的字符集的名称。它独立于 **charset** 字符串变量。调用 **metamail** 时，它也会复制到 **MM_CHARSET** 环境变量中。缺省值是 **US-ASCII** 。

easyeditor 内置编辑器 **~e** 命令的编辑器的名称。另请参阅 **editor** 字符串变量。缺省值为无。

editor 创建新邮件时使用的编辑器的名称。如果 **EDITOR** 环境变量已设置且不为空，则缺省值为该变量的值；否则为 **/usr/bin/vi** 。

可以使用 **none** 或 **builtin** 来指定内置编辑器。内置编辑器可用于在发送缓冲区中尚无文本的所有外发邮件（无转发的邮件、无回复中的复制消息、无任意外发消息中的签名等）。如果在发送缓冲区中有文本并且已指定 **builtin** ，则将使用由 **alteditor** 变量定义的编辑器。

另请参阅 **alteditor** 、 **easyeditor** 和 **visualeditor** 字符串变量。

escape 内置编辑器中使用的转义符。缺省值是波浪符 (~) 。

folderperms 邮件文件夹或 **aliases.text** 文件（由用户使用 **elm** 创建）的权限可以通过设置 **$HOME/.elm/elmrc** 文件中的 **folderperms** 值来进行配置。如果 **elm** 用户未设置值或未设置有效值， **elm** 将以 0640 作为缺省权限。 **elm** 向所有者授予读写权限。因此，如果 **folderperms** 在 **elmrc** 文件中的值不给所有者授予读写权限，则将使用缺省值来设置权限。

fullname

您发送邮件时将使用的邮件程序的名称。缺省值是您在 **/etc/passwd** 文件的条目中的 **pw_gecos** 字段的完整名称部分（第一个逗号之前的所有内容）。该字段可以使用 **chfn** 命令来设置（请参阅 *chfn*(1)、*finger*(1) 和 *passwd*(4)）。

localsignature

在调用编辑器之前自动追加到向本地主机发送的外发邮件的签名文件。它通常包含有关发件人的个人数据。另请参阅 **remotesignature** 字符串变量。缺省值为无。

收件人**:** 标头中的所有地址必须明显地指向本地主机。本地地址是在任何 **elm** 别名转换后不包含域名的地址。即，它们仅包含用户名（如 **santaclaus**）或者包含用户名和本地主机名（如 **santaclaus@northpole** ）。

即使 **santaclaus@northpole.arcticsea.org** 指向本地主机，仍将其视作远程地址。如果由 **sendmail** 系统别名列表重新指定地址的用户名满足上述条件，则将其当作本地地址。

maildir

邮件目录，通常在其中存储已接收和外发邮件的文件夹。缺省值是 **$HOME//Mail** 。

在 **elm ,** 中，可以使用 **=** 元字符指定该目录。例如，如果将消息保存到文件 **=/archive** ，则 **=** 将展开到 **maildir** 的当前值（斜线 (**/**) 是可选的）。

如果不存在 **maildir** 指定的目录，那么当启动 **elm** 时，系统将询问是否要将其创建。如果回答 **y** （是），则将创建该目录，并且将访问权限设置为 **700**。

pager

用于显示每条消息的程序。如果 **PAGER** 环境变量已设置且不为空，则缺省值为该变量的值；否则为内置寻呼 **builtin+** 。

通过内置寻呼 **builtin+** ，还可以在查看消息时执行一些“消息菜单”命令，内置寻呼包括一些简单的前进和后退滚动命令。当它处于活动状态时，输入 **?** 可获取命令的列表。另一种方法是使用 **more** 实用程序。

precedences

可以使用“标头菜单”放入外发邮件 优先级**:** 标头条目的优先权值的列表。每个优先权值可以选择性地与优先级值配对，后者自动放入 优先级**:** 标头条目中，从而将收到的消息标记为紧急。缺省值为无。

HP-UX 邮件传输代理， **sendmail** 可识别该标头。如果优先权值由 **sendmail** 配置文件 **/etc/mail/sendmail.cf** 中的 **P** 控制行定义，则将相应地调整消息的传输优先级。请参阅 *sendmail*(1M) 。

该条目的格式如下

precedences = *precedence*[**:***priority*] [*precedence*[**:***priority*]] ...

precedence 是优先权名称。 **/etc/mail/sendmail.cf** 中定义的缺省列表为:

first-class	传输优先级 0（缺省值）
special-delivery	传输优先级 100
list	传输优先级 -30
bulk	传输优先级 -60
junk	传输优先级 -100

priority 是放置在 优先级**:** 标头条目中的任意字符串。

prefix
外发消息中包含行的前缀。回复消息或者向他人转发消息时，可以选择性地包含初始消息。该前缀标记包含行。缺省值为 **>_** （ _ 被解释为空格字符）。

print
从各个菜单中执行 **p** （打印）命令时所运行的命令。该字符串有两种可能的格式：如果字符串包含特殊变量 **%s** ，则该变量将替换为包含消息的临时文件的名称，而命令将通过由 **shell** 字符串变量定义的 Shell 来执行。如果字符串不包含 **%s** ，临时文件名将追加到字符串，并且执行该命令。缺省值是

cat %s | lp

receivedmail
要将已接收邮件保存到的文件。缺省值是 **=received** ，即在由 **maildir** 定义的目录中接收到 (**received**) 的文件。

remotesignature
在调用编辑器之前自动追加到向远程主机发送的所有外发邮件的签名文件。它通常包含有关发件人的个人数据。另请参阅 **localsignature** 字符串变量。缺省值为无。

如果 收件人**:** 标头条目中的任何地址不是本地地址（如 **localsignature** 字符串所述），则将附加远程签名文件。

savecharset
用于在文件夹中保存消息的字符集。可能值为 **JIS** 、 **SJIS** 和 **EUC** 。如果未指定值，则将按照您的语言环境（由 **LC_TYPE** 和/或 **LANG** 环境变量指定）来保存消息。该选项仅适用于日语语言环境。缺省值为无。另请参阅 **jisconversion** 布尔变量。

sentmail
可在其中保存外发邮件副本的文件。一种可能性是收件箱 **/var/mail/***login-name* 。缺省值是 **=sent** ，即在由 **maildir** 定义的目录中发送 (**sent**) 的文件。

有关其他信息，请参阅 **copy** 布尔变量。

shell
用于 **!** 转义和其他类似操作的 Shell。如果 **SHELL** 环境变量已设置且不为空，则缺省值为该变量的值；否则为 **/usr/bin/ksh** 。

sortby
当前文件夹中索引的排序方式。可用选项包括:

from 发件人的名称。

sent 消息的发送日期。

received 消息的接收日期。

subject 消息的主题。将忽略前导 答复**:** （以及其他前导字符），因此回复将与初始消息一起进行排序。

lines 消息的行数。

status 阅读状态：空白、 **O** 和 **N** 。

可以给这些值添加 **reverse-** 前缀，以反转排序顺序。该值可以在“选项菜单”上修改。缺省值是 **reverse-sent** 。

textencoding 放入 MIME 内容**-**传输**-**编码**:** 标头条目的编码的类型。其选项是 **7bit** 或 **8bit**。缺省值是 **7bit** 。

tmpdir 临时文件的创建位置。如果 **TMPDIR** 环境变量已设置且不为空，则缺省值为该变量的值；否则为 **/tmp/** 。

visualeditor 用于内置编辑器 **~v** 命令的编辑器的名称。如果 **VISUAL** 环境变量已设置且不为空，则缺省值为该变量的值；否则为 **/usr/bin/vi** 。

weedout 阅读邮件时不需要显示的标头条目初始字符串的列表。通过将 **weed** 布尔变量设置为 **ON** ，可激活该列表。

该列表可以根据需要延续多行，前提是连续行均带有前导空白字符。要在字符串中包含空白字符，请用引号 (**"**) 将其引起来。您指定的字符串通常会追加到缺省列表，即：

>发件人
Apparently-To:
内容**-**长度
内容**-**传输**-**编码
内容**-**类型**:**
发件人
答复至**:**
MIME-Version
电子邮件程序**:**
Message-Id:
Newsgroups:
接收**:**
References:
状态**:**

X-Mailer:

有两个特殊值:

clear-weed-list

清除缺省列表。从 **weedout** 列表中删除缺省标头，这样您可以完全自定义列表。

end-of-user-headers

在任何后继行可能被误认为是列表中标头的情况下，标记 **weedout** 列表的末尾。

weedout 的缺省值是 ***end-of-user-headers*** 。

下划线 (_) 字符可用于指定空格。

请注意， 发件人会清除 发件人和 发件人**:** 。例如，如果要清除 发件人但不想清除 发件人**:** ，请指定 ***clear-weed-list*** ，其后接 **From_** 和不需要显示的其他任意标头。

数值变量

数值变量格式如下

numeric-name **=** *numeric-value*

下面定义下列数值变量。

bounceback
远程 UUCP 消息返回副本的阈值。如果目标主机超过离本地主机的指定跳越（网关）数，则目标主机将在收到消息时给您发送消息的副本。如果该值为 **0** ，则将禁用此功能。缺省值是 **0** 。

builtinlines
确定是否即使经常使用外部寻呼（在 **pager** 字符串变量中定义），仍应该对某些消息使用内置寻呼。可以通过两种方式定义是否应使用内置寻呼。

- 如果要对少于 *n* 行的所有消息使用内置寻呼，请将该值设置为 *n* 。
- 如果要对比屏幕行数少 *m* 行的所有消息使用内置寻呼，请将该值设置为 **-***m* 。

如果将该值设置为 **0** ，将始终通过外部寻呼发送消息。缺省值是 **-3** 。

noencoding
它可用于在邮件传输代理不支持 **8BITMIME** 和 **-B8BITMIME** 选项时发送原始的 8 位或二进制数据。缺省值是 **0** 。

可能值包括:

0 在发送前始终将 8 位和二进制消息转换为 7 位。

1 将 8 位消息转换为 7 位，但依赖于 **sendmail** 来处理二进制消息。

2 依赖于 **sendmail** 来处理 8 位和二进制消息。

readmsginc 读取新文件夹时 **Reading in** *folder*, **message:** 计数器的递增值。如果将该值设置为大于一的数字，则将加快在使用慢速终端读取大文件夹时的速度。缺省值是 **1** 。

sleepmsg 显示诊断消息后， **elm** 在清除该消息之前所等待的时间（秒钟）。该值可以是 **0** 或正整数。缺省值是 **2** 。

timeout **elm** 在收件箱中再次查找新邮件的时间间隔（秒钟）。缺省值是 **600** （10 分钟）。

userlevel 用户的相对经验程度。可接收的值包括：

0 入门用户（缺省值）。命令菜单是可用命令中较小的详细子集。

1 具有适当经验的用户。命令菜单是可用命令中较大的简洁子集。通过外发消息命令，可以将先前未发送的消息恢复为当前外发消息的文本。

2 专家级。其功能与 **1** 相同。

如果要发送表单消息，则需要具备级别 **1** 或 **2** 。

布尔变量

布尔变量格式如下

boolean-name **= ON** –和– *boolean-name* **= OFF**

下面定义下列布尔变量。

alwaysdelete 如果 **ON** ，“消息菜单” **q** （退出）命令的缺省答案将提示

是否删除邮件**? (y/n)**

设置为 **y** （是）。如果 **OFF** ，则缺省答案设置为 **n** （否）。缺省值是 **OFF**。请参阅“消息菜单” **q** 命令。

alwayskeep 如果 **ON** ，“消息菜单” **q** （退出）命令的缺省答案将提示

是否将未读邮件保留在收件邮箱中**? (y/n)**

设置为 **y** （是）。如果 **OFF** ，则缺省答案设置为 **n** （否）。缺省值是 **ON**。请参阅“消息菜单” **q** 命令。

alwaysstore 如果 **ON** ，“消息菜单” **q** （退出）命令的缺省答案将提示

是否将已读邮件移至 **received"** 文件夹中**? (y/n)"**

设置为 **y** （是）。如果 **OFF** ，则缺省答案设置为 **n** （否）。缺省值是 **OFF**。请参阅“消息菜单” **q** 命令。

arrow
如果 **ON** ，将使用箭头 **(->)** 在菜单索引中标记当前项目。如果 **OFF** ，将使用逆向条。如果使用 **-a** 命令行选项调用程序，则 **arrow** 设置为 **ON** 。缺省值是 **OFF** 。

ask
如果 **ON** ，系统将在适当情况下向您询问以下问题

是否删除邮件**? (y/n)**
是否将已读邮件移至 **received"** 文件夹中**? (y/n)"**
是否将未读邮件保留在收件邮箱中**? (y/n)**

- 每当您使用“消息菜单” **q** （退出）命令退出程序时。有关进程的详细信息，请参阅该命令。如果 **OFF** 或者使用“消息菜单” **Q** 命令， **elm** 将分别使用由 **alwaysdelete** 、 **alwaysstore** 和 **alwayskeep** 布尔变量定义的值，而不进行提示。缺省值是 **ON** 。

askcc
如果 **ON** ，每当您发送、转发或回复消息时， **elm** 都将用 抄送**:** 提示符提示您进行“抄送”。如果 **OFF** ，则将省略该提示符。在任一情况下，仍可以使用内置编辑器中的 **~c** 命令或使用“标头菜单”命令显式地包括 抄送**:** 地址。缺省值是 **ON** 。

autocopy
如果 **ON** ， **elm** 会自动将所回复的消息的文本复制到编辑缓冲区。如果 **OFF** ， **elm** 将提示您 是否复制邮件**?** 。缺省值是 **OFF** 。

confirmappend
如果 **ON** ，系统将在消息追加到现有文件之前要求您进行确认，而不管它是邮件目录中的文件还是其他目录中的文件。如果 **OFF** ，请参阅下面的 **confirmappend** 和 **confirmfiles** 操作。缺省值是 **OFF** 。

confirmcreate
如果 **ON** ，系统将在创建新文件之前要求您进行确认，而不管它是邮件目录中的文件还是其他目录中的文件。如果 **OFF** ，请参阅下面的 **confirmcreate** 和 **confirmfolders** 操作。缺省值是 **OFF** 。

confirmfiles
如果 **ON** ，系统将在消息追加到邮件目录之外其他目录中的现有文件之前要求您进行确认。它不影响邮件目录中的文件。如果 **OFF** ，请参阅下面的 **confirmappend** 和 **confirmfiles** 操作。缺省值是 **OFF** 。

confirmfolders
如果 **ON** ，系统将在从邮件目录中创建新文件之前要求您进行确认。它不影响其他目录中的文件。如果 **OFF** ，请参阅下面的 **confirmcreate** 和 **confirmfolders** 操作。缺省值是 **OFF** 。

confirmcreate 和 **confirmfolders** 操作

confirmcreate=ON 和 **confirmfolders=ON**
在邮件目录中创建文件之前进行确认。在其他目录中创建文件之前进行确认。

ON 和 **OFF** 在邮件目录中创建文件之前进行确认。在其他目录中创建文件之前进行确认。

OFF 和 **ON** 在邮件目录中创建文件之前进行确认。不在其他目录中创建文件之前进行确认。

OFF 和 **OFF** 不在邮件目录中创建文件之前进行确认。不在其他目录中创建文件之前进行确认。

confirmappend 和 **confirmfiles** 操作

confirmappend=ON 和 **confirmfiles=ON**
在追加到邮件目录中的文件之前进行确认。在追加到其他目录中的文件之前进行确认。

ON 和 **OFF** 在追加到邮件目录中的文件之前进行确认。在追加到其他目录中的文件之前进行确认。

OFF 和 **ON** 在追加到邮件目录中的文件之前进行确认。不在追加到其他目录中的文件之前进行确认。

OFF 和 **OFF** 不在追加到邮件目录中的文件之前进行确认。不在追加到其他目录中的文件之前进行确认。

copy 如果 **ON** ，则将在外发步骤保存所有外发邮件的静默副本。如果 **OFF** ，则不保存副本。缺省值是 **OFF** 。

如果 **ON** 并且 **savename** 布尔变量为 **ON** ， **elm** 首先会尝试将其保存到按照 **savename** 的定义命名的文件。如果存在该文件，则将保存消息。如果不存在该文件，但 **forcename** 布尔变量为 **ON** ，则将创建该文件并保存消息。如果 **forcename=OFF** ，消息将保存到由 **sentmail** 字符串变量定义的文件。如果 **savename=OFF** ，消息将保存到由 **sentmail** 字符串变量定义的文件。

forcename 如果 **ON** ，则即使文件夹不存在，当按收件人登录名保存外发消息时，仍将创建该文件夹。如果 **OFF** ，则不创建文件夹。缺省值是 **OFF** 。

有关其他信息，请参阅 **copy** 布尔变量。

forms 如果为 **ON** ，并且 **userlevel** 数值变量是 **1** 或 **2** ，则可以创建表单消息。“发送菜单” **m** （创建表单）命令会将您的消息转换为表单消息。如果 **OFF**，则不能创建表单消息。缺省值是 **ON** 。

jisconversion 如果 **ON**，则将在发送之前将外发邮件转换为 JIS（日本行业标准）。如果 **OFF**，则不进行转换。该选项仅适用于日语语言环境、 **ja_JP.SJIS** 和 **ja_JP.eucJP**。缺省值是 **OFF**. **savecharset** 字符串变量。

keepempty 如果 **ON**，则将保留已删除其中所有消息的文件夹。如果 **OFF**，则删除空文件夹。缺省值是 **OFF**。

keypad 如果 **ON**，则启用 HP 2622 终端光标键。如果 **OFF**，则禁用光标键。如果使用 **-K** 命令行选项调用程序，则 **keypad** 设置为 **OFF**。另请参阅 **softkeys** 布尔变量。缺省值是 **ON**。

menu 如果 **OFF**，则禁止所有程序屏幕显示上的菜单显示。如果 **ON**，将显示菜单。如果使用 **-m** 命令行选项调用程序，则 **menu** 设置为 **OFF**。缺省值是 **ON**。

metoo 如果 **ON**，将给您发送您向也包含您的名称的别名发送的消息的副本。如果 **OFF**，则不发送该副本。缺省值是 **OFF**。

mimeforward 如果 **ON**，转发的消息将作为附件发送。如果 **OFF**，则转发的消息将作为主消息的一部分发送。缺省值是 **ON**。

movepage 如果 **ON**，在邮箱中按页移动的命令（ **+** 和 **-** 命令）也会将当前索引指针移到指向新消息页的顶部。如果 **OFF**，则按页移动命令不会更改当前消息指针位置。缺省值是 **ON**。

names 如果 **ON**，在展开外发消息的 收件人**:** 别名时仅显示用户名。如果 OFF，则显示整个展开地址。缺省值是 **ON**。

nohdrencoding 如果 **ON**，则不对包含 8 位或多字节字符的标头值执行 RFC 1522 编码。如果 **OFF**，则执行编码。缺省值是 **OFF**。

noheader 如果 **ON**，在为回复或转发将消息复制到文件缓冲区时将不包括消息的标头。如果 **OFF**，则复制所有标头。缺省值是 **ON**。

noheaderfwd 如果 **ON**，在为转发将消息复制到文件缓冲区时将不包括标头。如果 **OFF**，则复制所有标头。对于转发，该选项将覆盖 **noheader** 的设置。缺省值是 **OFF**。

pagemultipart 如果 **ON**，则使用 **pager** 变量的值显示包含未知子部分或未知子类型的 MIME 多部分消息。如果 **OFF**，则调用 **metamail** 来查看多部分消息。缺省值是 **OFF**。

pointnew 如果 **ON**，则在启动时自动指向消息索引中的第一条新消息。如果 **OFF**，则指向第一条消息。在任一情况下，如果启动文件夹不是收件箱，或者没有新消息，均将指向第一条消息。缺省值是 **ON**。

promptafter 如果 **ON** ，则将在外部寻呼退出后显示命令提示。如果 **OFF** ，则返回到调用菜单。缺省值是 **ON** 。

resolve 如果 **ON** ，在删除、取消删除或转发消息后，会将指针移到指向索引中的第一条消息。如果 **OFF** ，则将指针保留在当前消息处。缺省值是 **ON** 。

savename 如果 **ON** 并且正在保存消息， **elm** 将在 **maildir** 目录中利用消息发件人的用户名构建建议的文件名，其格式为 **=***username* 。如果 **OFF** ，则不建议任何文件名。

如果 **ON** 并且正在发送将保存的消息， **elm** 将基于 收件人**:** 列表中第一个条目的用户名构建文件名，其格式与上面的格式相同。如果 **OFF** ，则不构建任何文件名。有关其他信息，请参阅 **copy** 布尔变量。

缺省值是 **ON** 。

sigdashes 如果 **ON** ，则在本地或远程签名文件中所包括的签名文本之上插入两个短线。这是通用的惯例。如果 **OFF** ，则忽略短线。缺省值是 **ON** 。

softkeys 如果 **ON** ，则启用 HP 2622 终端功能键协议。如果 **OFF** ，则禁用该功能键协议。如果使用 **-k** 或 **-K** 命令行选项调用程序，则 **softkeys** 设置为 **OFF** 。另请参阅 **keypad** 布尔变量。缺省值是 **OFF** 。

titles 如果 **ON** ，则用以下形式的行给显示的消息添加标题：

Message *number***/***total sendername date time*

sendername 、 *date* 和 *time* 按“消息索引”所述的方式从消息标头中提取。这在使用 **weedout** 列表禁止相关标头条目时非常有用。如果 **OFF** ，则不给消息添加标题。缺省值是 **ON** 。

usetite 如果 **ON** ，则使用 **termcap ti/te** 和 **terminfo cup** 光标定位条目（请参阅 *terminfo*(4) ）。如果 **OFF** ，则不使用这些条目。如果使用 **-t** 命令行选项调用程序，则 **usetite** 设置为 **OFF** 。缺省值是 **ON** 。

weed 如果 **ON** ，在显示消息供阅读时，将不显示由 **weedout** 变量定义的标头。如果 **OFF** ，则显示所有标头。缺省值是 **ON** 。

METAMAIL 配置

MIME（多用途 Internet 邮件扩展）编码按照内容传输编码（用于使消息可通过邮件传输的编码）和内容类型（消息部分在解码后的类型和形式）将消息及其附件进行分类。“附件配置菜单”小节和 RFC 1521 将详细介绍编码和类型。

elm 为以下内容类型提供内置的支持：

text/plain [**;** **charset=***charset*]

text 是可显示字符集 *charset* （缺省为 **US-ASCII** ）中的全部字符。

multipart/mixed **;** **boundary=***boundary-string*

消息由大量单独的“正文部分”组成，它们用 **--***boundary-string* 进行分隔，每一部分包含定义内容类型和内容传输编码的可选标头。缺省内容类型是 **text/plain** 。

multipart/digest **;** **boundary=***boundary-string*

它与 **multipart/mixed** 类似，例外的是缺省内容类型为 **message/rfc822** 。

multipart/report **;** **boundary=***boundary-string*

message/rfc822

该消息由另一个标准消息格式的消息组成。

metamail 是一个系统程序，由 **elm** 调用来管理无法在普通 ASCII 文本中显示的消息和附件的显示。

metamail 提供对其他内容类型的外部支持，这些内容类型在一个或多个 **mailcap** 文件中定义。系统 **mailcap** 文件为 **/etc/mail/mailcap** 。可以在 **$HOME/.mailcap** 中自定义缺省的 **mailcap** 文件。也可以通过设置 **MAILCAPS** 环境变量来指定您自己的 **mailcap** 文件列表。按顺序搜索 **mailcap** 文件，直至找到匹配内容类型和任何条件的条目。

最少程度的 **mailcap** 条目由如下形式的行组成：

content-type **;** *command*

command 是您为查看指定内容类型的文件而键入的命令，而字符串 **%s** 替换为文件名。例如，要查看原本是 HTML 源文本且内容类型为 **text/html** 的正文部分，可以使用以下条目

text/html; netscape %s

同样，对于 GIF 图像文件，可以使用以下条目

image/gif; xv %s

RFC 1521 定义了大量的内容类型， **elm** 将这些内容类型交给 **metamail** 来处理：

text/richtext
multipart/alternative
multipart/parallel
multipart/digest
message/partial
message/external-body
image/jpeg

image/gif
audio/basic
video/mpeg
application/octet-stream
application/postscript

在系统 **mailcap** 文件中查找处理多个内容类型的条目。

外部语言环境影响

环境变量

HOME 您的主（登录）目录。

EDITOR 如果已设置且不为空，则提供 **alteditor** 和 **editor** 字符串变量的缺省值。

LANG 如果已设置且不为空，则确定显示消息所使用的语言。缺省值是 **C** 。请参阅 *environ*(5) 。

MAILCAPS 如果已设置，则定义 **metamail** 使用的 **mailcap** 文件的搜索路径缺省值是

$HOME/.mailcap:/etc/mail/mailcap

PAGER 如果已设置且不为空，则提供 **pager** 字符串变量的缺省值。

SHELL 如果已设置且不为空，则提供 **shell** 字符串变量的缺省值。

TMPDIR 如果已设置且不为空，则提供 **tmpdir** 字符串变量的缺省值。

VISUAL 如果已设置且不为空，则提供 **visualeditor** 字符串变量的缺省值。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

消息模式示例

要在不加载主 **elm** 邮件处理程序的情况下发送邮件，请使用由程序名后接收件人登录和可选地址组成的简单命令格式。 **elm** 将提示指定主题和副本，然后启动一个编辑器，用于撰写消息（用户响应为斜体）：

$ *elm j_doe*
收件人**: doe (John Doe)**
主题**:** *this is a test*
抄送**:** ...
... 调用编辑器，撰写消息，然后...
Your options now are:
a)ttachments e)dit message edit h)eader s)end it f)orget it.

What is your choice? *s*
mail sent!

如果“忘记”了消息，它保存在 **$HOME/Canceled.mail** 中。

包含重定向的文件模式

要使用命令行重定向发送文件，请使用如下命令：

$ elm j_doe < help.c

它读取文件 **help.c** 并用缺省主题发送。

包含管道的文件模式

通过邮件发送命令输出并包括命令行：

$ ls -a | elm -s "Directory Listing" j_doe

警告

如果使用两个不同的邮件程序同时访问相同的邮件文件（通常是从两个不同的窗口中意外地进行访问），则可能导致无法预料的结果。

主题字符串（与 **-s** 命令行选项一起使用）和文件名（与 **-f** 和 **-i** 命令行选项一起使用）的长度不能超过 255 个字符。如果长度超过该限制，则会将字符串截断到 255 个字符。

作者

elm 由 HP 开发。

文件

$HOME/.elm	用户的 **elm** 别名、配置、标头和其他文件的目录
$HOME/.elm/aliases	用户别名数据库数据表
$HOME/.elm/aliases.dir	用户别名数据库目录表
$HOME/.elm/aliases.pag	用户别名数据库散列表
$HOME/.elm/aliases.text	用户别名源文本
$HOME/.elm/elmheaders	用户定义的附加标头
$HOME/.elm/elmrc	用户配置文件
$HOME/Canceled.mail	在非交互式使用中取消的消息。
/tmp/alias.*pid*	用于删除别名的临时文件
/tmp/form.*pid*	用于表单消息的编辑器缓冲区
/tmp/mbox.*loginname*	用于用户 *logname* 的临时邮箱
/tmp/print.*pid*	用于输出消息的临时文件
/tmp/snd.*pid*	外发邮件消息编辑缓冲区
/tmp/sndh.*pid*	外发邮件标头编辑缓冲区
/usr/lib/nls/msg/C/elm.cat	消息目录的位置
/usr/share/lib/elm/elmrc-info	**$HOME/.elm/elmrc** 文件的注释文件
/var/mail	接收邮件的目录； 它必须具有模式 **755** 和组 ID **mail**

/var/mail/.elm	**elm** 邮件程序系统别名的目录
/var/mail/.elm/aliases	系统别名数据库数据表
/var/mail/.elm/aliases.dir	系统别名数据库目录表
/var/mail/.elm/aliases.pag	系统别名数据库散列表
/var/mail/.elm/aliases.text	系统别名源文本
/var/mail/*loginname*	用户的收件箱; 它必须具有模式 **660** 和组 ID **mail**
/var/mail/*loginname***.lock**	邮件目录锁

另请参阅

answer(1)、 chfn(1)、 elmalias(1)、 fastmail(1)、 finger(1)、 mailfrom(1)、 newalias(1)、 newmail(1)、 readmail(1)、 vi(1)、 sendmail(1M)、 passwd(4)、 terminfo(4)、 environ(5)。

RFC 821 “简单邮件传输协议 (SMTP)”

RFC 822 “Internet 文本消息格式标准”

RFC 1341 “MIME（多用途 Internet 邮件扩展）：用于指定和描述 Internet 消息正文格式的机制”

RFC 1521 “MIME（多用途 Internet 邮件扩展）第一部分：用于指定和描述 Internet 消息正文格式的机制”

RFC 1522 “MIME（多用途 Internet 邮件扩展）第二部分：非 ASCII 文本的消息标头扩展”

名称

elmalias - 显示和验证 elm 用户和系统别名

概要

elmalias [**-dersu**] [**-a**|**-f** *format*|**-n**|**-v**|**-V**] [*alias-name-list*]

备注

newalias 命令接管了 **elmalias** 命令以前的功能（请参阅 *newalias*(1)）。

说明

elmalias 命令显示和验证用户和系统 **elm** 别名。

系统数据库必须已由 **newalias** 命令创建（请参阅 *newalias*(1)）。用户数据库必须已由 **newalias** 命令或 **elm** 邮件系统创建（请参阅 *elm*(1)）。如果这两个数据库具有相同的别名，则使用用户版本。缺失的数据库文件将被忽略，并且不会有任何提示。

每个数据库条目可以具有下列字段，在 *newalias*(1) 中对下列字段进行了详细说明：

alias-list 条目的一个或多个别名列表。

address-list 条目的一个或多个地址列表。地址可以是其他条目的 *alias-list* 中的别名。

comment 包含关于条目信息的可选字段。在出站邮件中不包含此字段。

firstname 一个可选字段，被视为人名或组名。它用于 *fullname* 中。

lastname 一个可选字段，被视为人员或组的姓。它用于 *fullname* 中。

fullname 由 *firstname* 字段和 *lastname* 字段组成的组合值，其形式为： *firstname lastname* 。

elmalias 识别三种类型的别名：

人员 在 *address-list* 中具有一个地址的数据库条目。 **elmalias** 假定此地址是一个有效的邮件地址。

组 *address-list* 中具有两个或多个地址的数据库条目。 **elmalias** 最初假定这些地址是 人员条目或 组条目的别名。

未知 组条目中的地址或 *alias-name-list* 中的别名，而不是数据库中的别名。在这两种情况下，将该项作为别名和别名地址来报告。

在不具有选项或操作数的情况下， **elmalias** 显示两个数据库中每个别名的 *address-list* 字段。如果一个条目具有多个别名，则多次显示 *address-list* 字段。

具有 *alias-name-list* 而不具有选项的情况下， **elmalias** 显示列表中每个别名的 *address-list* 字段。如果在数据库中未查找到别名，则将其视为 未知，不加注释。

选项

elmalias 可识别下列选项：

-a 将显示更改为别名，其后紧跟 *address-list* 字段。

-d 启用调试。

-e 完全扩展 组别名。只有给定 *alias-name-list* 时，才能使用此选项。

如果 组地址列表中的地址是别名，则该地址由该别名条目替换。该进程是递归的，直到结果为 人员类型或 未知类型为止。如果 组地址不是别名，则以 未知类型，将该地址既作为别名又作为地址进行报告。重复的别名只报告一次。 人员条目不被扩展，即使这些条目的地址实际为别名，亦如此。

-f *format* 显示文件中或 *alias-name-list* 中每个别名的 *format* 字符串。 *format* 中的下列字符对由相应的每个别名值替换。

%a 别名。

%c *comment* 字段。

%l *lastname* 字段。

%n *fullname* 值。

%t 别名类型： 人员、 组或 未知。

%v *address-list* 字段。

-n 将显示更改为 *address-list* ，其后紧跟 *fullname* ，如果有全名，则放在括号中。

-r 如果 *alias-name-list* 中的名称与数据库中的别名不对应，则报告错误。显示每个未知名称的消息，然后以非零状态退出。

-s 仅使用系统别名数据库，除非还指定了 **-u** 。

-u 仅使用用户别名数据库，除非还指定了 **-s** 。

-v 使用详细输出格式。将显示更改为别名，别名后紧跟 *address-list* ，地址列表后紧跟 *fullname* ，如果有全名，则放在括号中。

-V 使用非常详细的多行输出格式，该格式具有下列标题，与 **-f** 选项的格式代码相对应。如果字段为空，则省略标题。

别名**:**
地址**:**
类型**:**
名称**:**
Last Name:
注释**:**

退出状态

elmalias 设置下列退出状态值:

0 正常完成。

<>0 发生错误。您也许指定了:

+ 无效选项。
- **-e** 选项不具有 *alias-name-list* 。
- **-f** 选项不具有 *format* 。
- **-r** 选项在 *alias-name-list* 中具有未知别名。

举例

请考虑包含下列条目的用户数据库:

```
# sample alias file
mom = My Mommy, Work: x2468 = my_mother@a.computer
dad,father,pop = Father; Dear, Work: x1357 = host!otherhost!dad
parents = The Folks = mom dad parent@host
siblings = The Kids = brother1
  brother2
  sister
brother1 = Son; First = bro1@kid.computer
```

由于 **brother2** 和 **sister** 没有指向别名条目，因此，将它们作为 未知键入。

不具有选项或操作数的 **elmalias** 生成下列输出结果。

```
my_mother@a.computer
host!otherhost!dad
host!otherhost!dad
host!otherhost!dad
mom,dad,parent@host
brother1,brother2,sister
bro1@kid.computer
```

elmalias -v 生成别名、地址和全名，如下所示:

```
mom        my_mother@a.computer (My Mommy)
dad        host!otherhost!dad (Dear Father)
father     host!otherhost!dad (Dear Father)
pop        host!otherhost!dad (Dear Father)
parents    mom,dad,parent@host (The Folks)
siblings   brother1,brother2,sister (The Kids)
brother1   bro1@kid.computer (First Son)
```

要扩展一组别名，并用字段标题对其进行格式化，请使用 **-e** 选项和 **-f** 选项，如下命令中所示：

elmalias -ef "Alias: %a Address: %v Type: %t" parents siblings

生成：

别名**: mom** 地址**: my_mother@a.computer** 类型**:** 人员
别名**: dad** 地址**: host!otherhost!dad** 类型**:** 人员
别名**: parent@host** 地址**: parent@host** 类型**:** 未知
别名**: brother1** 地址**: bro1@kid.computer** 类型**:** 人员
别名**: brother2** 地址**: brother2** 类型**:** 未知
别名**: sister** 地址**: sister** 类型**:** 未知

作者

elmalias 由 HP 开发。

文件

$HOME/.elm/aliases	用户别名数据库数据表
$HOME/.elm/aliases.dir	用户别名数据库目录表
$HOME/.elm/aliases.pag	用户别名数据库哈希表
$HOME/.elm/aliases.text	用户别名源文本
/var/mail/.elm/aliases	系统别名数据库数据表
/var/mail/.elm/aliases.dir	系统别名数据库目录表
/var/mail/.elm/aliases.pag	系统别名数据库哈希表
/var/mail/.elm/aliases.text	系统别名源文本

另请参阅

elm(1)、 newalias(1)。

名称

enable、disable - 启用/禁用 LP 打印机

概要

enable *printers*

disable [**-c**] [**-r**[*reason*]] *printers*

说明

enable 命令可激活指定的 *printers* ，使其打印 **lp** 接收的请求。可用 **lpstat** 查询打印机的状态（请参阅 *lp*(1) 以及 *lpstat*(1) ）。

disable 可停用指定的 *printers* ，使其无法打印 **lp** 接收的请求。缺省情况下，在指定的打印机上当前正在打印的任何请求，会在同一台打印机或相同类型的其他打印机上完整地重新打印。可用 **lpstat** 查询打印机状态。

选项

disable 可识别下列选项：

-c 取消任何指定打印机上当前正在打印的请求。

-r[*reason*] 将一个 *reason* 与停用打印机相关联。此原因适用于在下一个 **-r** 选项之前提及的所有打印机。如果不存在 **-r** 选项或没有为 **-r** 选项指定原因，则将使用缺省 *reason* 。 *reason* 由 **lpstat** 报告。 *reason* 的最大长度为 80 个字节。长度大于 80 个字节的 *reason* 将被截断至 80 个字节。

外部语言环境影响

环境变量

LANG 用于确定显示消息的语言。

如果不指定 **LANG** ，或者其值为空，则会缺省为 **C** （请参阅 *lang*(5) ）。

如果任一国际化变量包含无效设置，则所有国际化变量将缺省为 **C** （请参阅 *environ*(5) ）。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

启用打印机 **snowwhite** 以接收请求：

enable snowwhite

停用打印机 **snowwhite** 并取消任何已登录的作业：

disable -c snowwhite

警告

如果启用了限制取消功能（由 **lpadmin -orc** 选项选择 — 请参阅 *lpadmin*(1M) ），且用户既不是管理员，也不是指定打印机上当前打印请求的所有者，则 **disable** 将忽略 **-c** 选项。

enable 和 **disable** 仅在本地系统上执行各自的操作。

文件

/etc/lp/*
/usr/lib/lp/*
/var/adm/lp/*
/var/spool/lp/*

另请参阅

lp(1)、 lpstat(1)、 accept(1M)、 lpadmin(1M)、 lpsched(1M)、 rcancel(1M)、 rlp(1M)、 rlpdaemon(1M)、 rlpstat(1M)。

名称

env - 设置命令执行的环境

概要

env [**-**] [**-i**] [*name* **=** *value*] ... [*command* [*arguments* ...]]

说明

env 获取当前 *environment* ，并根据其参数修改该环境，然后在修改后的环境下执行命令。在执行命令前，*name***=***value* 形式的参数会合并到继承的环境中。 **-i** 选项会完全忽略继承的环境，使得命令完全在参数指定的环境下执行。 **-** 选项已过时，它与 **-i** 选项具有相同的作用。

如果不指定命令，则输出生成的环境，每行输出一个名称-值对。

返回值

如果调用了 *command* ，则 **env** 的退出状态是 *command* 的退出状态；否则， **env** 退出时返回下列值之一：

- **0** **env** 成功完成。
- **1-125** **env** 遇到错误。
- **126** 找到 *command* 但无法调用。
- **127** 找不到 *command* 。

外部语言环境影响

环境变量

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_MESSAGES** 或将其设为空字符串，则 **LANG** 的值将用作每个未指定或空变量的缺省值。如果不指定 **LANG** 或将其设为空字符串，则使用缺省的“C”（请参阅 *lang*(5) ）而非 **LANG**。

如果任一国际化变量包含无效设置，则 **env** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

警告

- 选项已过时。可使用 **-i** 取代。

另请参阅

sh(1)、 exec(2)、 profile(4)、 environ(5)。

符合的标准

env：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

eucset - 设置和获取 ldterm 的代码宽度

概要

eucset [**-p**]

eucset [[**-c** *HP15-codeset*] 或 [**-c** *UTF8*] 或 [**-c** *GB18030*] 或 [*cswidth*]]

说明

eucset 命令设置或者获取（报告）由当前输入终端处理的扩展 UNIX 代码 (EUC)、UCS 转换格式 (UTF8) 和 GB18030 字符的编码和显示宽度。 EUC 是一种编码方法，适用于由单字节或多字节组成的代码集。它允许应用程序和终端硬件使用 7 位 US ASCII 代码，并允许同时使用多达三个单字节或多字节的代码集。

不带任何选项的 **eucset** 命令首先尝试将代码集设置为四个 HP15 代码集之一。如果失败，则使用 7 位 US ASCII 作为缺省的代码集。必须使用该命令指定任何其他 EUC 代码集，无论它们是单字节的还是多字节的。有关 *cswidth* 参数值的特殊警告信息，请参阅“警告”一节。

对于 GB18030 或 UTF8 设置，请使用 **-c** 选项。

选项

eucset 命令可识别下列选项和参数：

-p 显示终端的 EUC 字符宽度的当前设置。

-c 将宽度设置为四个 HP15 代码集、 **UTF8** 或 **GB18030** 其中之一。支持的 HP15 代码集包括 **SJIS** 、 **CCDC** 、 **GB** 和 **BIG5** 。

EUC 代码集类

EUC 将代码集分成四类。每个代码集有两个特性：用于对代码集中的字符进行编码的字节数，以及用于显示代码集中字符的显示列数。代码集内的所有字符具有相同的特性。

- 代码集 0 由所有 7 位单字节 ASCII 字符组成。这些字符中每个字符的最重要的位是 0（零）。代码集 0 中的字符需要一个字节用于编码，并占用一个显示列。这些值对代码集 0（零）而言是固定的。 7 位 US ASCII 代码是主要的 EUC 代码集，用户无需直接指定就可以使用它。
- 代码集 1 是补充的 EUC 代码集。代码集 1 的字符具有一个初始字节，它的最重要的位是 1。代码集 1 内的字符可能需要多个字节用于编码，并且可能需要多个显示列。必须使用 **eucset** 命令设置代码集 1 的特性。
- 代码集 2 和 3 是补充的 EUC 代码集。这两个代码集中字符的初始字节分别是 SS2 或 SS3。它们需要多个字节用于编码，并且可能需要多个显示列。必须使用 **eucset** 命令设置代码集 2 和 3 的特性。

eucset 命令行中的 *cswidth* 参数是一个字符串，它说明代码集 1 至 3 的字符宽度。该命令不允许用户修改代码集 0 的设置。字符串采用以下格式：

X1 [:*Y1*],*X2* [:*Y2*],*X3* [:*Y3*]

值 *X1* 表示对代码集类 1 中的字符进行编码所需的字节数。 *Y1* 表示显示该类中的字符所需的显示列数。 *X2* 表示

对代码集 2 中的字符进行编码所需的字节数，不算 SS2 字节， *Y2* 表示代码集 2 中字符的显示列数。 *X3* 表示对代码集 3 中的字符进行编码所需的字节数，不算 SS3 字节， *Y3* 表示这些字符所需的显示列数。如果列宽度值等于进行编码的字节数，则可能忽略列宽度值。如果将任意 EUC 代码集的编码值设置为 **0** （零），则表示代码集不存在。有关 *cswidth* 参数值的特殊警告信息，请参阅“警告”一节。

如果没有提供 *cswidth* 参数， **eucset** 命令将使用 **CSWIDTH** 环境变量的值。如果没有提供该变量，则替换以下缺省字符串：

1:1,0:0,0:0

该缺省字符串指定环境使用单字节 EUC 代码集，它具有 EUC 代码集 1 格式的字符。如果环境使用代码集 1 格式的多字节 EUC 代码集、代码集 2 或 3 格式的单字节或多字节 EUC 代码集（或二者兼有），则不能使用缺省设置。

外部语言环境影响

环境变量

LANG 提供未设置的或空的国际化变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值 **C** （请参阅 *lang*(5) ），而不是使用 **LANG** 。如果任一国际化变量包含无效设置，则 **eucset** 就会认为所有国际化变量都设置为 **C** 。请参阅 *environ*(5) 。

LC_ALL 如果设置为非空字符串值，则覆盖所有其他国际化变量的值。

LC_MESSAGES 确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息以及写入到标准输出的信息消息的格式和内容。

NLSPATH 确定消息清单的位置，以便处理 **LC_MESSAGES** 。

举例

要显示环境中 EUC 代码集 1 至 3 的编码和显示宽度，请输入：

eucset -p

假定先前已经使用 **eucset** 针对 **ja_JP.eucJP** 进行了设置，则该条目将生成以下信息：

cswidth 2:2,1:1,2:2

要将代码集 1 和 2 中 EUC 字符的编码和显示宽度的当前设置，各自更改为两个字节，请输入下列内容之一：

eucset 2:2,2:2,0:0

eucset 2,2,0

要设置语言环境 **ja_JP.eucJP** 中 EUC 字符的编码和显示宽度，请输入：

eucset 2:2,1:1,2:2

对于 **zh_TW.eucTW** ，请输入：

eucset 2:2,3:2

对于 **ko_KR.eucKR** ，请输入：

eucset 2:2

要将代码宽度设置为 **UTF8** 的代码宽度，请输入：

eucset -c UTF8

要将代码宽度设置为 **GB18030** 的代码宽度，请输入：

eucset -c GB18030

警告

cswidth 参数在字节宽度值中不包括 SS2 或 SS3 字节。

该命令不是由标准指定的，在其他供应商的系统上可能未提供该命令，并且在将来的版本中可能会更改或过时。

作者

eucset 由 OSF 和 HP 开发。

另请参阅

dtterm(1)、 ldterm(1)。

名称

ex、edit - 扩展的、面向行的文本编辑器

概要

ex [**-**] [**-l**] [**-r**] [**-R**] [**-t***tag*] [**-v**] [**-w***size*] [**-x**] [**-C**] [**+***command*] [*file*]...

XPG4 概要

ex [**-rR**] [**-s**|**-v**] [**-c***command*] [**-t***tag*] [**-w***size*] [*file*]...

过时形式选项

ex [**-rR**] [**-**|**-v**] [+*command*] [**-t***tag*] [**-w***size*] [*file*]...

edit [**-**] [**-l**] [**-r**] [**-R**] [**-t***tag*] [**-v**] [**-w***size*] [**-x**] [**-C**] [**+***command*] [*file*]...

备注

程序名 **ex** 、 **edit** 、 **vi** 、 **view** 和 **vedit** 表示同一程序的不同形式。本联机帮助页描述 **ex/edit** 的行为。在许多 HP-UX 系统和其他类似系统上， **e** 是 **ex** 的同义词。

说明

ex 是面向行的文本编辑器，它还支持面向屏幕的编辑（请参阅 *vi*(1) ）。

（仅适用于 XPG4）。某些块模式的终端不能够支持完整的 **ex** 定义，如全屏幕编辑命令（**visual** 模式或 **open** 模式）。当此类终端不支持这些命令时，将发出一个“not an editor command”错误消息，或者报告一个语法错误。

edit 程序与 **ex** 是等同的，只是更改了一些缺省的编辑器选项，以方便初学者和临时用户使用。（请参阅下面的编辑器选项）。

选项和参数

ex 可识别下列命令行选项和参数:

- （已过时）禁止所有交互式用户的反馈。当编辑器命令来自于脚本时，这是很有用的。

-s （仅适用于 XPG4）。

禁止所有交互式用户的反馈。当编辑器命令来自于脚本时，这是有用的。

忽略 **TERM** 的值以及任何实现终端的类型，并假设终端不支持可视模式。

禁止使用 **EXINIT** 环境变量，并禁止读取 **.exrc** 文件。

-l 设置 **lisp** 编辑器选项（请参阅下文的编辑器选项）。

-r 在编辑器或系统崩溃之后，恢复指定的 *file* 。如果没有指定的 *file* ，则输出所有已保存的文件的清单。要恢复已保存的文件，用户必须是这个文件的所有者。（超级用户不能恢复其他用户的文件）。

-R 设置 **readonly** 编辑器选项，以防止意外覆盖文件。（请参阅下面的编辑器选项）。

-t *tag* （仅适用于 XPG4）。编辑包含指定 *tag* 的文件，并将第一个命令视为 **:tag** *tag* 。 **-t** *tag* 和 **ta** 所表示的命令是可选的。如果系统提供了 **ctags** 的实现，那么应提供这个命令，否则使用 **-t** 将产

生不可预测的结果。

执行 **tag** *tag* 命令，以加载并定位一个预定义文件。请参阅“命令描述”中的 **tag** 命令和 **tags** 编辑器选项，后者在下面的编辑器选项一节。

-v 调用可视模式 (**vi**) 。

-w *size* 将 **window** 编辑器选项的值设置成为 *size* （请参阅下面的编辑器选项）。如果省略了 *size* ，则缺省值是 **3** 。

-x 设置加密模式。提示用户输入一个密钥，以开始创建一个加密文件或编辑一个已加密的文件（请参阅下面的“命令描述“一节中的 **crypt** 命令）。

-C 加密选项。与 **-x** 选项相同，但是它假设已读取的所有文本都已经加密。

-c *command* （仅适用于 XPG4）。
+*command* （已过时）通过执行指定的 **ex** 搜索或定位 *command* 来开始编辑。

file 指定需要编辑的一个或多个文件。如果指定了多个 *file* ，将根据给定的顺序来处理文件。如果还指定了 **-r** 选项，将从恢复区读取文件。

（仅适用于 XPG4）。如果同时指定了 **-t** *tag* 和 **-c** *command* 选项，则首先处理 **-t** *tag* 。例如，先通过 **-t** 来选择包含了标记的文件，然后执行命令。

定义

“当前文件” 。 **ex** 正在编辑的文件称为当前文件。将当前文件中的文本读取至工作区，所有编辑修改都是在工作区执行的。将工作区的内容显式地回写至一个文件之前，修改并不会影响原始文件。如果将 **%** 字符用作文件名，则用当前文件名替换它。

“备用文件” 。备用文件是编辑器命令中提及的最后一个文件。或者，如果最后一个文件成为当前文件，那么前一个当前文件成为备用文件。如果 **#** 字符用作文件名，将使用备用文件的文件名替换。

“缓冲区” 。在编辑过程中，可以使用名称为 **a** 到 **z** 的 26 个缓冲区。如果缓冲区名称是大写的，那么文本会追加到缓冲区，而不是覆盖。

“只读标志” 。可以在编辑器内清除 **readonly** 标识，这一步可以通过在编辑器选项内设置 **noreadonly** 来完成（请参阅下面的编辑器选项）。即使设置了 **readonly** 标志，也允许将文本写入不同的文件。此外还可以将文本强制写入 **readonly** 文件，这可以通过在 write 命令后使用 **!** 来实现（请参阅下面的“命令描述”中的 **write** 命令）。

“中断” 。如果收到中断信号和键盘输入的命令，那么 **ex** 返回命令模式。如果编辑器命令来源于文件，中断信号将导致 **ex** 异常中止。

“系统崩溃” 。如果系统崩溃，或者因为内部错误或意外信号而导致 **ex** 异常中止但是又没有保存所做的修改，那么 **ex** 将试图保存工作区的内容。使用 **-r** 命令行选项可以读取已保存的修改内容。

“命令模式/输入模式”。 **ex** 以命令模式启动，并以冒号 (**:**) 提示符开头。如果遇到 **append** 、 **change** 或 **insert** 命令， **ex** 切换至输入模式。若欲终止输入模式并返回到命令模式，可以在行的开头输入句点 (**.**) 。

“注释” 。以引号 (") 开头的命令行都将被忽略（这对于在编辑器脚本中放置注释是很有用的）。

“多重命令”通过使用竖线字符 (|) 分隔，可以在一行组合多个命令。但是全局命令、注释和 Shell 转义命令必须位于行的末尾，因为它们不能使用 | 字符来结尾。

地址

（仅适用于 XPG4）。 **ex** 中的地址与当前行有关。一般说来，当前行是命令所能影响的最后一行。对当前行的确切影响将在每个命令的描述中讨论。当缓冲区不包含任何行时，将当前行设置为零。

ex 可识别以下形式的行地址:

. 点或句点 (**.**) 表示当前行。通常当前行的位置可以是执行明确的移动命令的结果，或者是作用于多个行的命令的结果（在此情况下，当前行通常是受影响的最后一行）。

n 工作区中的第 *n* 行。行的编号是连续的，从第 1 行开始。

$ 工作区的最后一行。

% **1,$** 的缩写，表示整个工作区。

+*n*，**+[+]**...
-*n*，**-[-]**... 相对于当前行或前面一行规范的偏移量。 **+** 表示向前； **-** 表示向后。例如， **.+3** 、 **+3** 和 **+++** 是等同的。

/*re***/**
?*re***?** 行包含 *re* 模式，向前扫描 (**/**) 或向后扫描 (**?**). 。如果仅显示行，则可以省略前导 **/** 或 **?**。如果省略了 *re* ， **ex** 使用最近用到的一组扫描字符串或替换字符串（请参阅下面的“正则表达式”一节）。

’*x* 用单个小写字母标记的行（请参阅下面“命令描述”一节中的 **mark** 命令）。 **’***x* 表示用 *x* 标记的行。此外，每次在进行非相对移动之前，将标记前一当前行。使用 **’** 来代替 *x*，可以表示此行（因此 **’’** 表示前一当前行）。

（仅适用于 XPG4）。命令需要零个、一个或两个地址。对于需要零个地址的命令，如果出现地址，命令将地址视为错误。

（仅适用于 XPG4）。对于一个 **range** 内的相邻地址，应使用逗号 (**,**) 或分号 (**;**) 分隔。对于后一种情况，应将当前行 (**.**) 设置成为首地址，然后才能计算第二个地址。这一特性可以用来确定向前搜索和向后搜索的的启始行。在任何包含两个地址的序列中，第二个地址对应于第一个地址。第一个地址应小于或等于第二个地址。第一个地址应大于或等于编辑缓冲区中启始行的地址，最后一个地址应小于或等于编辑缓冲区中最后一行的地址，否则将产生错误。

命令的地址由一系列的行地址组成（如上所述），行地址用逗号 (**,**) 或分号 (**;**). 分隔。此类地址列表是从左向右计算的。当分隔符是分号时，在解释下一个地址之前将前一地址的值设置成为当前行。如果给定地址的数量超过了命令所要求的数量，那么除了最后一个或两个地址，其他所有地址都会忽略。如果命令需要两个地址，那么在工作区中，给定了地址的首行应位于第二行之前。缺省方式下，地址列表中的空（缺失）地址给予当前行。

正则表达式

无论何时，编辑器总是维护两个正则表达式字符串的副本：这两个字符串是替换字符串和扫描字符串。替换命令将替换字符串设置为使用的正则表达式。全局命令和所有命令的行地址的正则表达示形式（请参阅上面的“地址”一节）都将扫描字符串设置为使用的正则表达式。如“地址”一节所述，缺省情况下，这些字符串用作正则表达式，即 **global** 命令以及 **substitute** 命令。

编辑器支持基本正则表达式（请参阅 *regexp*(5) ），但有下列修改：

\< — **\<** 与“单词”的开头部分相匹配；也就是说，相匹配的字符串必须以字母、数字或下划线开头，而且其前面只能是行的开头部分和或与上述不相同的字符。这种结构只能用于正则表达式的开头（如 \<*word* 中），而不能用于中间 (*word1* \<*word2*) 。

\> — **\>** 与“单词”的末尾相匹配（请参阅前面的段落）。这种结构只能用于正则表达式的末尾（如 *word*\> 中），但是不能用在中间 (*word1*\> *word2*) 。

~ — 匹配最后一个 **substitute** 命令中的替换部分。

[*string***]** — 通过使用反斜杠 (\) 引用括号表达式的特殊字符，替换由基本正则表达式定义的括号表达式内的位置引用。

nomagic — 当设置了 **nomagic** 编辑器选项时，具有特殊含义的字符仅仅是模式开头处的 ^ 和模式末尾的 **$** 以及 \。字符 “.”、“*”、“ **[** ”和“ ~ ”将失去它们的特殊含义，除非使用 **** 转义。

替换字符串

字符 **&** 表示需要替换的模式所匹配的文本。如果设置了 **nomagic** 编辑器选项，则使用 **\\&** 。

字符 ~ 会用前一个 **substitute** 命令的替换部分替换。如果设置了 **nomagic** 编辑器选项，则使用 **\\~** 。

序列 *****n*，（如果 *n* 是一个整数）将用与子模式匹配的文本替换，子模式在第 *n* 组 **\\(** 和 **\\)** 括号中。

序列 **\\u** (**\\l**) 后面的字符如果是字母，此序列将字符转换成小写字母。序列 **\\U** (**\\L**) 打开大小写转换，直到遇到序列 **\\E** 或 **\\e** ，或到达替换字符串的末尾。

命令名称和缩写

下表总结了行模式的命令。括号中的命令只有缩写形式。

命令	缩写	命令	缩写	命令	缩写
abbreviate	***ab***	***next***	***n***	***tag***	***ta***
append	***a***	***number***	***nu #***	***unabbreviate***	***una***
args	***ar***	***open***	***o***	***undo***	***u***
change	***c***	***pop***		***unmap***	***unm***
chdir	***chd cd***	***preserve***	***pre***	***version***	***ve***
copy	***co t***	***print***	***p***	***visual***	***vi***
crypt	***cr X***	***put***	***pu***	***write***	***w wq***
delete	***d***	***quit***	***q***	***xit***	***x***
edit	***e ex***	***read***	***r***	***yank***	***ya***
file	***f***	***recover***	***rec***		
global	***g v***	***rewind***	***rew***	(execute buffer)	*** @**
insert	***i***	***set***	***se***	(line number)	**=**
join	***j***	***shell***	***sh***	(left shift)	**<**
list	***l***	***source***	***so***	(right shift)	**>**
map		***stop***	***st ^Z***	(scroll)	***^D***
mark	***ma k***	***substitute***	***s sr & ~***	(shell escape)	***!***
move	***m***	***suspend***	***su ^Z***	(window)	***z***

命令描述

在下面的命令描述中，有些参数是经常出现的。下面对它们进行了描述。

line 单行地址，形式如前文“地址”一节中所述。缺省为当前行。

range 由逗号或分号分开的行地址对，请参阅上面的“地址”一节。缺省是当前行 (**.,.**) 。

count 一个正整数，它指定命令所作用的行数。缺省值是 1 或 *range* 中指定的行数。

指定 *count* 后， *range* 将不起作用。相反，只应指定一个行号，以说明命令所作用的第一行。（如果给定了范围，则将范围的最后一行指定为命令的启始行）。

flags 一个或多个 **#** 、 **p** 和 **l** 字符。命令结束后，将执行相应的命令以输出行。可以使用这些标志来指定任意数量的 **+** 或 **-** 字符。缺省是没有任何标志。

这些修饰符都是可选的。

仅仅（用空命令）指定 *line* 或 *range* 时，隐含的命令是 **print** 。如果输入一个空行，将输出下一行（等同于 **.+1p**）。

buffer “**XPG4** 功能”。多个命名区中的一个，用于保存文本。命名缓冲区是在 POSIX 语言环境中用小写字母指定的。指定 **buffer** 应导致受到命令作用的文本区会存储到缓冲区, 而事实上却还没有执行相关的存储命令。这个参数还用于 **put** 命令和可视模式的“put”命令 （**p** 和 **P**），以指定提供插入文本的缓冲区。

如果缓冲区名称为大写，而且需要修改此缓冲区，那么应在其后追加文本，而不能覆盖它。如果不需要修改缓冲区，那么缓冲区名称可以为小写，也可以为大写，其结果是相同。还有一个未命令缓冲区，当没有指定缓冲区时，它用作删除或移出的所有文本的储备库。

此外还有编号缓冲区，编号为 1 到 9。只有在可视模下才能访问它们。这些缓冲区的特殊之处在于，在可视模式下，当被删除的文本放在未命名缓冲区时，它还会放在缓冲区 1，而前一缓冲区中的内容应放在缓冲区 2，依此类推。缓冲区 9 中的文本会丢失。移到未命名缓冲区中的文本不应修改编号缓冲区。不能直接将文本放到编号缓冲区，尽管可以使用可视模式的“put”命令和编号式的缓冲区名称来检索文本。当后面的命令没有使用缓冲区修饰符时，缺省为未命名缓冲区。

word “**XPG4** 功能”。在 POSIX 语言环境中， **word** 由最大的字母、数字和下划线序列组成，其两端用非字母、数字或下划线的字符分隔，也可以用单词或文件的开头和结尾分隔。 **!** 此字符可以追加至命令，以修改命令的运行。后面的单个命令描述进行详细介绍。

如果同时为命令指定了 **count** 和 **range** ，那么将根据 **count** 的值而不是根据 **range** 的值来确定作用的行数。命令的启始行将作用于范围指定的第一行。

当仅仅指定了 **line** 或 **range** 而没有指定命令时，暗含的命令将是 **print** 、 **list** 或 **number** (**p** 、 **l** 或 **#**) 。选中的命令应该是这三个命令中的最后一个命令。如果没有指定范围和行数，而且命令行是一个空行，那么应写入当前行，而且当前行应设置为 **.+1** 。

地址、行数、命令名称的前面或后面可以是零或 <空格> 字符。如果命令名称后面的对象（如缓冲区和文件等）以字母字符开头，那么至少要用一个 <空格> 将它与命令分开。

对于下面列出的命令，可以使用缩写来输入（“摘要”中的命令单词以 [开头），也可以使用完整命令（省略了 [和] 的命令名称）或完整命令的以缩写开头的任何子字符集。

abbreviate **ab**[**breviate**] *word replacement*

向当前行添加命名缩写。在可视模式下，如果在输入过程中， *word* 是作为一个完整的单词输入的，将用字符串 *replacement* 替换它。

append *line* **a**[**ppend**][**!**]

进入输入模式，文本放在指定行的后面。如果指定了第 0 行，文本放在工作区的开头。如果没有输入任何行，最后输入行将成为当前行或目标行。

向命令追加 **!**，仅仅为本次插入切换 **autoindent** 编辑器选项设置。

args **ar**[**gs**]

输出参数，将当前参数放入 **[** 和 **]** 之间。

change *range* **c**[**hange**][**!**] *count*

进入输入模式，用输入文本替换指定的行。上一输入行成为当前行；如果没有输入任何行，那么作用与删除相同。

向命令追加 **!**，仅仅为本次插入切换 **autoindent** 编辑器选项。

chdir **chd**[**ir**][**!**] [*directory*]

cd[**!**] [*directory*]

将工作目录更改为 *directory* 。如果省略了 *directory* ，将使用 **HOME** 环境变量的值。如果上次写入之后修改了工作区，而且正在编辑的文件的名称不是以斜线 (/) 开头的，将发出一个警告，而且不改变工作目录。在这种情况下，若要强制改变目录，应在命令中追加字符 **!** 。

copy *range* **co**[**py**] *line flags*

range **t** *line flags*

将指定的多个行 (*range*) 的副本放在指定目标 *line* 的后面；第 0 行指定了需要放在工作区开头的行。（字符 **t** 是 **copy** 命令的另一缩写）。

crypt **cr**[**ypt**]

X

提示用户输入密钥，并用此密钥进入加密模式。此命令还可用于更改前一 **crypt** 命令或 **-x** 命令行输入的密钥。如果没有提供密钥（即仅输入了回车键），那么将取消加密模式，后面的写入命令以纯文格式操作工作区。

在加密模式下，使用当前密钥加密所有输入文件。但是在处理输入文件时，如果遇到一个文本块（约 1024 字节），而且这个文本块仅包含 7 位的 ASCII 字符，将认为它是纯文本的，而且不加密它。除了通过 **!** Shell 转义管道输出至另一命令的文件输出，其他所有文件输出都使用当前密钥加密。

编辑器为管理工作区而使用的临时文件不加密，直到忽略（或写出）当前工作区并开始编辑新文件。当创建需要加密保护的新文件时，请确保在调用编辑器时通过指定 **-x** 选项加密了工作区文件。

cr[**ypt**]

C

加密选项。除了假定读取的所有文本都已经加密以外，其他与 **X** 相同。

delete *range* **d**[**elete**] *buffer count*

从工作区删除指定行。如果指定了命名的 *buffer* ，将在这个缓冲区中保存删除的文本。如果没有指定缓冲区，则使用未命名缓冲区（即存放最近被删除或移出文本的缺省缓冲区）。删

除行的后一行成为当前行，如果删除行位于文件的末尾，则文件的最后一行成为当前行。

edit **e**[**dit**][**!**] [**+** *line*] *file*
ex[**!**] [**+** *line*] *file*

开始编辑新文件。 (**ex** 是命令 **edit** 的另一名称。如果上次写入操作之后修改了当前工作区，则发出一个警告，命令异常中止。将符号 **!** 追加给命令 (**e!** *file*) 可以覆盖这个操作。当前行是工作区的最后一行，除非在 *vi* 内执行本命令（在这种情况下，当前行是工作区的第一行）。如果指定了 **+***line* 选项，将当前行设置为指定位置， *line* 可以是一个数字（或 **$** ）或指定为 **/***re* 或 **?***re* 。

file **f**[**ile**]

输出当前文件的文件名和其他信息，包括行数和当前位置。

global *range* **g**[**lobal**][**!**] **/***re***/** *command*...
range **v /***re***/** *command*...

在包含了 *re* 的 *range* 内的行上执行命令 *command* （如果没有指定 *range* ，则在整个工作区上执行命令）。首先标记在给定 *range* 内与模式 *re* 相匹配的行。如果省略了模式，将使用最近的替换字符串或扫描字符串（请参阅前面的“正则表达式”）。然后执行给定的 *command*，同时将 **.** 设置为标识行。可以用除字母和数字以外的任何字符来分隔模式，以替代 **/** 。

通过使用反斜杠隐藏换行符的方式， *command* 可以在多行指定。如果省略了 *command* ，将输出所有行。允许使用 **append** 、 **change** 和 **insert** 命令；如果结束点用来结束 *command* 或 *command* ，则不能省略。还允许使用 **visual** 命令（除非 **global** 命令本身由可视模式发出），也可以从终端读取输入。（如果 *command* 包含一个可视模式命令（如 **open** 或 **visual**），必须用可视模式的 **Q** 命令来终止可视模式命令，以处理下一个标记行）。

在 *command* 中禁止使用 **global** 命令本身和 **undo** 命令。禁止使用编辑器选项 **autoprint** 、 **autoindent** 和 **report** 。

将一个 **!** 字符追加给 **global** 命令（即 **g!** ）或使用另一替代名称 **v** 将导致 *command* 运行在 *range* 范围内与模式不匹配的行上。

insert *line* **i**[**nsert**][**!**]

进入输入模式；输入的文本放在指定行之前。最后一行输入成为当前行；如果没有输入任何行，则目标行的前一行成为当前行。

向命令追加 **!** ，仅仅为本次插入切换 **autoindent** 编辑器选项设置。

join *range* **j**[**oin**][**!**] *count flags*

将指定行中的文本连接到另一行。调整空格部分，提供不少于一个的空格字符（如果行的末尾是句点，则提供两个空格字符；如果行的首个字符在封闭的括号 ()) 中，则不提供空格字

符）。行开头部分多余的空格将会忽略。

将 **!** 追加给命令进行简单的连接而不处理空格。

list *range* **l[ist]** *ount flags*

输出指定行，制表符显示为 **^I** ，标记行的末尾显示为 **$** （唯一有用的标志是行号的 **#** 标志）。最后输出行成为当前行。

map **map** key|**#***n action*
map! key|**#***n action*

map 和 **map!** 定义的宏用于可视模式。第一个参数 *key* 可以是单个字符，也可以是多字符序列。特殊序列 **#***n*, 中的 *n* 是表示功能键 *n* 的数字。在参数中输入特殊字符、空格和换行符时，必须先使用 **^V** 转义。 *key* 参数的第一个字符不能是冒号 (**:**) ，而多字符序列不能以字母字符开头。

map 定义的宏用于可视命令模式。 **map!** 定义的宏用于可视输入模式。当输入与 **#***n* 相应的 *key* 或功能键时，编辑器将操作解释为输入了相应的 *action* 。

不带选项的 **map** 或 **map!** 命令将显示相应的当前宏列表。

另请参阅下面的“命令描述”一节中的 **keyboardedit** 、 **keyboardedit!** 、 **timeout** 和 **timeoutlen**

编辑器选项。

mark *line* **ma[rk]** *x*
line **k** *x*

为指定行设置指定标记 *x* ，这个标记必须是 (**a-z**) 中的一个小写字母。 x 前面必须是空格或制表符。对当前行位置没有影响。 **k mark** 是另一替代名称。

move *range* **m[ove]** *line*

将指定的多个行 (*range*) 移到目标 *line* 后面。移动的第一行成为当前行。

next **n[ext][!]** [*file*]...

编辑命令行参数中的下一个文件。向命令追加 **!** 将覆盖警告信息，这些警告信息提示上次写入操作之后修改了工作区（而且，除非设置了 **autowrite** 编辑器选项，任何修改都会忽略）。可以通过在此命令行上指定一个新的参数列表来替换这个参数列表。

number *range* **nu[mber]** *count flags*
range **#** *count flags*

（ **#** 字符是 **number** 命令的另一缩写）。输出行，每个行的前面都是行号（唯一有用的标志是 **l** ）。最后一个输出行成为当前行。

open *line* **o**[**pen**] **/***re***/** *flags*

进入开放模式，此模式与单行窗口模式相似。可以使用所有可视模式命令。如果在 *line* 中发现与可选正则表达式相匹配的部分，光标将位于匹配模式的开头。使用可视模式命令 **Q** 退出开放模式。有关详细信息，请参阅 *vi*(1) 。

pop **pop**[**!**]

加载文件名存储在标记栈顶的文件，并将当前行设置为存储位置。删除栈顶条目。（当执行行模式的 **tag** 命令或可视模式的 **^]** 命令时，当前文件名放在栈中）。

! 覆盖警告信息，这些警告信息提示此写入操作之后会修改工作区；而且，除非设置了 **autowrite** 编辑器选项，任何修改都会忽略）。

preserve **pre**[**serve**]

如果系统崩溃，则保存当前编辑器工作区。此命令适用于紧急情况，例如，写入操作失效，而且用任何方法都不能保存工作区。使用 **-r** 命令行选项恢复文件。保存文件后，向用户发送邮件消息。消息应包含文件名、保存时间和一个可用来恢复文件的 **ex** 命令。邮件消息还应包含其他附加信息。

print **range p**[**rint**] *count*

输出指定行，将非打印字符输出成 **^x** 形式的控制字符；DEL用 **^?** 表示。最后一个输出行成为当前行。

put *line* **pu**[**t**] *buffer*

将删除行或“移出行”放在 *line* 之后。可以指定缓冲区；否则，将在未命名缓冲区（即存储删除或移出文本的缺省缓冲区）中存储文本。应将回置的第一行设置成为当前行。

quit **q**[**uit**][**!**]

终止编辑。如果上次写入操作之后修改了工作区，将输出一个警告消息且命令失败。若要强制终止编辑而不保存修改，应向命令追加 **!** 。

read *line* **r**[**ead**] *file*

将工作区中指定 *file* 的副本放在目标行后面（若要把文本放在开头，可以把目标行设置为 0 行）。如果没有命名 *file* ，缺省文件是当前文件。如果没有当前文件， *file* 成为当前文件。除了在可视模式下读取的第一行成为当前行的情况，读取的最后一行都成为当前行。

如果用 **file** 来指定 **!***string*, *string* ，那么会将其作为系统命令解释并将它传递给命令解释器；输出结果读入工作区。 **!** 前面必须是空格或制表符。

recover **rec**[**over**][**!**] *file*

发生意外挂起或系统崩溃后， *file* 从保存区恢复文件。如果上次写入操作后修改了工作区，将输出一个警告信息，随后命令异常中止。通过向命令追加 **!** (**rec!** *file*) ，可以覆盖这项操

作。

rewind **rew**[**ind**][**!**]

倒回参数列表，并编辑列表中的第一个文件。此命令等同于操作数是当前参数列表的 **next** 命令。如果上次写入操作后修改了当前缓冲区，将输出一个警告信息，并且命令异常中止。通过追加 **!** 可以覆盖警告信息。当前指示器行应受到编辑器选项 **autowrite** 和 **writeany** 的影响。

set **se**[**t**] [**all**]
se[**t**] [**no**]*boolean-option***?**
se[**t**] *value-option*[**?**]
se[**t**] *boolean-option*
se[**t**] **no***boolean-option*
se[**t**] *value-option***=***value*

设置并显示编辑器选项的值（请参阅下面的编辑器选项）。

如果没有参数，将输出值修改为缺省设置的编辑器选项。如果指定了 **all** ，那么输出所有当前选项值。

第二种和第三种形式显示指定选项的当前值。仅 Boolean 选项需要 **?** 。

第四种形式打开 Boolean 选项。第五种形式关闭 Boolean 选项。

第六种形式将值分配给字符串选项和数字选项。必须使用 (\) 前缀转义字符串中的空格和制表符。

可以组合后五种形式；解释是从左向右进行的。

shell **sh**[**ell**]

执行 **shell** 编辑器选项指定的命令解释器（请参阅下面的编辑器选项）。退出命令解释器之后，将恢复编辑。

source **so**[**urce**] *file*

从指定的 *file* 读取并执行命令。 **so** 命令是可以嵌套的。最大嵌套深度由由实现定义，但至少应有一层嵌套。

substitute *range* **s**[**ubstitute**] **/***re***/***repl***/** *options count flags*
range **s** *options count flags*
range **&** *options count flags*
range **sr** *options count flags*
range **~** *options count flags*
range **s\?***repl*

range **s\&***repl*

在每个指定行上，用字符串 *repl* 替换模式 *re* 的第一个实例（请参阅上面的“正则表达式”和“替换字符串”）。可用任何除字母和数字以外的字符替换 **/** 来界定模式。

如果包含了 **g** （全局）选项，将替换行中模式的所有实例。

如果包含 **c** （确认）选项，将要求用户确认是否进行各个替换，过程如下：在每次替换之前，显示行和用符号 (^) 标记的需替换模式。键入 **y** 将进行替换；其他输入将导致异常中止。最后一个替换行成为当前行。

如果省略替换模式 *re* (**s//***repl***/**)，将使用最近的替换字符串组和扫描字符串组(请参阅前面的“正则表达式”)。

如果使用 **s** 或 **&** 命令形式，替换字符串缺省为前一替换字符串。

如果使用 **sr** 或 **~** 命令形式，替换模式缺省为最近使用的一组替换字符串或扫描字符串。代替字符串缺省为前一被使用的代替字符串。

s\?*repl* 形式与 **s/***scan-re***/***repl***/** 形式是等同的，这里 *scan-re* 是前一扫描字符串。

s\&*repl* 形式与 **s/***subs-re***/***repl***/** 形式是等同的，这里的 *subs-re* 是前一替换字符串。

suspend **su**[**spend**][**!**]

stop **st**[**op**][**!**]

susp

挂起编辑器作业并返回至调用 Shell。 **stop** 和 *susp* 与 **suspend** 是等同的。 *susp* 是用户进程控制挂起字符，一般用字符 **^Z** (ASCII SUB) 表示（请参阅 *stty*(1)）。如果调用 Shell 不支持或禁止了作业控制，此命令将禁用。

如果设置了 **autowrite** 编辑器选项，没有设置 **readonly** 编辑器选项，而且上次写入操作后修改了工作区，在编辑器挂起之前，将工作区的内容写入当前文件。若要覆盖此操作，应将字符 **!** 追加至 **suspend** 命令或 **stop** 命令。

tag **ta**[**g**][**!**] *tag*

依次搜索由 **tags** 编辑器选项指定的文件（请参阅下面的编辑器选项），直到查找到 *tag* 的标记定义。如果查找到 *tag* ，将相关文件加载到工作区，并将当前位置设置为标记定义中指定的地址。

如果新文件与当前文件不相同，而且设置了 **autowrite** 编辑器选项但没有设置 **readonly** 编辑器选项，并且上次写入操作后修改了工作区，那么在加载新文件之前，将工作区的内容写入新文件。若要覆盖此操作，应在命令后追加 **!** 字符。

如果设置了 **tagstack** 编辑器选项，那么将当前文件名和行号推入标记栈，供后面的 **pop** 命令或可视模式下的 **^]** 命令调用。

unabbreviate **una**[**bbreviate**] *word*

删除缩写列表中的 *word* （请参阅前面的 **abbreviate** ）。

undo **u**[**ndo**]

撤消前一编辑命令所作的修改。为此， **global** 和 **visual** 被视为两个单命令。不能撤消作用于外部环境的命令（如 **write** 、 **edit** 和 **next** ）。可以撤消 **undo** 命令本身。

unmap **unm**[**ap**][**!**] *key*

撤消为 *key* 定义的宏（请参阅前面的 **map** ）。

version **ve**[**rsion**]

输出编辑器的当前版本信息。

visual *line* **vi**[**sual**] *type count flags*

在指定的 *line* 进入可视模式。

与 **z** （窗口）命令中相同， *type* 可以是字符 **+** 、 **-** 、 **.** 或 ^ 中的一个字符，以指定屏幕上指定行的位置。

count 指定初始窗口的大小；缺省是 **window** 编辑器选项的值。

标志 **#** 和 **l** (ell) 将导致用相应的模式在可视窗口中显示行（请参阅 **number** 和 **list** 命令）。

使用 **Q** 命令可退出可视模式。有关详细信息，请参阅 *vi*(1) 。

write [*range*] **w**[**rite**][**!**][**>>**] *file*
[*range*] **wq**[**!**][**>>**] *file*

将指定行（如果没有指定 *range* ，则是整个工作区）写出至 *file* ，并输出行号和写出的字符。如果没有指定 *file* ，缺省是当前文件 (如果没有当前文件而且没有指定任何文件，命令失败，并输出错误消息。

如果指定了备用文件，而且存在这一备用文件，那么写入操作失败，但是可以通过向命令追加字符 **!** 来强制执行。若要追加至现有文件，应在命令后追加 **>>** 符号。如果文件不存在，则报告错误。

如果文件指定为 **!***string* ，那么 *string* 解释为系统命令，然后调用命令解释器，并将指定行作为标准输入传递给命令。

wq 命令等同于 **w** 后接一个 **q** 命令； **wq!** 命令等同于 **w!** 命令后接一个 **q** 命令； **wq>>** 命令等同于 **w>>** 命令后接 **q** 命令。

xit **x**[**it**][**!**][**>>**] *file*

如果工作区发生改变，将使用 **write** 命令所使用的任意选项（如 **!** ， **>>** 或 *file* ）来执行 **write** 命令。然后（无论何种情况）执行 **quit** 命令。

yank *range* **ya**[**nk**] *buffer count*

将指定行放在命名的 *buffer* 。如果没有指定缓冲区，将使用未命名缓冲区（即存放最近删除或移出文本的缺省缓冲区）。

(execute buffer) ***** [*buffer*]
@ [*buffer*]

将 *buffer* 的内容作为编辑器命令来执行。 *buffer* 可以是命名缓冲区 (**a–z**) 的字母，或者是 ***** 或 **@** 。此命令的 ***** 形式与 **@** 形式是等同的。如果没有指定缓冲区，或者 *buffer* 是 ***** 或 **@** ，将执行 ***** 或 **@** 命令中最后一个命名缓冲区。

(line number) *line* **=** *flags*

输出指定 *line* 的行号。缺省是最后一行。当前行位置不受影响。

(scroll) **^D**

输出后面的 *n* 行， *n* 是 **scroll** 编辑器选项的值。

(shell escape) **!** *command*
range **!** *command*

将 **!** 后面的部分传递给系统命令解释器执行。如果上次写入操作后修改了工作区，将发出警告信息。命令完成后输出一个 **!** 。当前行位置不受影响。

在 *command* 的文本内， **%** 和 **#** 被扩展为文件名，并用前一个 **!** 命令的文本替换 **!** 。因此， **!!** 重复前一个 **!** 命令。进行扩展后，将回显一个被扩展了的行。

如果指定了 *range* ，那么将指定行作为标准输入传递给命令解释器。 *command* 的输出替换指定行。

(shift left) *range* **<** *count flags*

将指定行向左转移。需要删除的空格由 **shiftwidth** 编辑器选项确定。在转移过程中，只丢弃空白字符（空格和制表符），其他字符不受影响。最后一个转移行成为当前行。

(shift right) *range* **>** *count flags*

通过插入空白字符,将指定行向右转移。插入的空白字符的个数由 **shiftwidth** 编辑器选项确定。最后一个转移行成为当前行。

(window) *line* **z** *type count flags*

显示由 *count* 指定的行。 *count* 缺省是 **window** 编辑器选项的值。

如果省略了 *type* ，将输出指定 *line* 后的 *count* 行。

如果指定了 *type* ，那么它必须是下列字符中其中之一:

+ 显示地址行后的一个行窗口。

- 将地址行放在一个显示行窗口的底部。

. 将地址行放在窗口的中心。

^ 显示与地址行相隔两个窗口的前一个行窗口。

= 在窗口中心显示地址行，并在其上下各显示一行短线。

对于 **=** ，地址行成为当前行，其他的都是最后一个输出行成为当前行。

编辑器选项

命令 **ex** 有多个可以修改其行为的选项。这些选项都有缺省设置，可以使用 **set** 命令来更改这些设置（请参阅前面的内容）。通过将 **set** 命令字符串加入到环境变量 **EXINIT** 、 **HOME** 目录下的 **.exrc** 文件或当前目录下的 **.exrc** 之中，可以在启动时设置选项。如果有 **EXINIT** ，将不执行 **HOME** 目录下的 **.exrc** 文件。如果当前目录不是 **HOME** 目录，而且没有设置 **exrc** 编辑器选项（请参阅下面的内容），那么将在执行 **EXINIT** 或 **HOME** 目录下的 **.exrc** 文件之后，执行当前目录下的 **.exrc** 。

编辑器根据 **terminfo** 数据库获得终端屏幕的大小（请参阅 *terminfo*(4) ）。 **UNIX95** 环境变量说明为本命令使用 XPG4 行为，使用 **COLUMNS** 和 **LINES** 环境变量可以覆盖这些值。更多相关信息请参阅下面的 **window** 编辑器选项。

下表是不同形式的编辑器的缺省设置:

名称	缺省编辑器选项				
edit	***nomagic***	***novice***	***noreadonly***	***report=1***	***showmode***
ex	***magic***	***nonovice***	***noreadonly***	***report=5***	***noshowmode***
vedit	***nomagic***	***novice***	***noreadonly***	***report=1***	***showmode***
vi	***magic***	***nonovice***	***noreadonly***	***report=5***	***noshowmode***
view	***magic***	***nonovice***	***readonly***	***report=5***	***noshowmode***

除非明确说明，编辑器选项都是 Boolean 类型的。括号内是它们的缩写。

autoindent (**ai**)
（使用空格和制表符）缩进输入模式中的每一行，使之与前一行对齐。在追加一行之后、插入一行或修改第一行之前，缩进开始进行。还可以提供其他的常规缩进操作。将自动缩进后续行以重新对齐。

一次或多次输入 **^D** 可以减少缩进操作：对于每次 **^D** ，光标向后移动并消除 **shiftwidth** 个空格位置。将在 ^ 加在 **^D** 之前将撤消当前行的所有临时缩进。将 **0** 加在 **^D** 之前将撤消所有缩进。

用 **noautoindent** (**noai**) 可以撤消它。缺省值为 **noautoindent** 。

autoprint (**ap**) 在命令改变工作区文本后输出当前行。 **global** 命令禁止使用 autoprint 命令。可以通过 **noautoprint** (**noap**) 撤消。缺省值为 **autoprint** 。

autowrite (**aw**) 如果修改了工作区，而且给定了 **next** 、 **rewind** 或 **!** ，将工作区写出至当前文件。可以使用 **noautowrite** (**noaw**) 来撤消。缺省值为 **noautowrite** 。

beautify (**bf**) 将输入文本中除制表符、换行符和换页符以外的所有控制符全部忽略。可以使用 **nobeautify** (**nobf**) 来撤消。缺省值为 **nobeautify** 。

directory=*dirname* (**dir**)

指定编辑器工作区的目录。只有创建工作区后，此选项才能生效。如果编辑文件是在命令行上指定的，那么必须在 **EXINIT** 或 **.exrc** 中设置此选项，才能用于编辑文件的工作区文件位置。缺省值为 **/var/tmp** 。

如果指定目录是在 **EXINIT** 或 **.exrc** 文件中设置的，而且用户不能写入此文件，那么编辑器将退出；如果用户交互式地设置，编辑器将报告出错信息。

doubleescape 设置后，需要连续两个 ESC（退出键）才能退出输入模式。在输入模式下，一个 ESC 后接一个其他不同字符将导致 **vi** 发出一个有声的或图示的警告（请参阅 **flash** 编辑器选项），并将这两个字符插入工作区。可以使用 **nodoubleescape** 来撤消它。缺省值为 **nodoubleescape** 。

通过某些终端的键盘的编辑键转换而成的字符序列与一些 **vi** 用户命令是等同的。如果启动了这些键的映射（请参阅 **keyboardedit** 和 **keyboardedit!** 选项）， **vi** 可能不能够可靠地区分由编辑键转换而成的字符序列和用户输入的相同字符序列。特别是当用户想要在一个 **vi** 命令前输入一个 ESC 以立即终止输入模式时，最容易发生此问题。如果设置了 **doubleescape** 选项，就不会出现这样的问题。

edcompatible (**ed**) 记忆替换命令上的 **g** 和 **c** 前缀，并通过重复这两个前缀来切换它们。可以通过 **noedcompatible** (**noed**) 来撤消它。缺省值为 **noedcompatible** 。

errorbells (**eb**) 设置此选项后，仅在不支持突出显示模式或高亮模式（如反显模式）的终端上用警报声报告出错消息。如果终端支持高亮模式，则不用警报声来报告出错信息，本选项无效。请注意，在可视模式下，总是用警报声和出错消息来报告出错（而不考虑这个选项的设置）。

可以用 **noerrorbells** (**noeb**) 来撤消它。缺省值为 **noerrorbells** 。

exrc 设置此选项后，如果当前目录不是 **HOME** 目录，那么在编辑器初始化过程中，处理当前目录下的 **.exrc** 文件。这个选项不能使用缺省值来设置，若要生效，必须在 **EXINIT** 环境变量和 **HOME** 目录下的 **.exrc** 文件中进行设置。请参阅前面的编辑器选项中的介绍。可以使用 **noexrc** 来撤消它。缺省值为 **noexrc** 。

flash (**fl**) 设置此选项后，如果在终端的 **/usr/share/lib/terminfo** 数据库中添加了 **flash_screen** 条目，屏幕将闪烁，而不是发出嘟嘟声。可以使用 **noflash** (**nofl**) 来撤消它。缺省值为 **flash** 。

hardtabs=*number* (**ht**)
在扩展制表符时，定义硬件制表符设置与系统使用的空格数之间的间格。在每个与 *number* 的整数倍相对应的列号（从屏幕的左边界开始）中放置制表位。缺省值为 **hardtabs=8** 。

ignorecase (**ic**)
正则表达式匹配中，将文本的所有大写与小写相映射。此外，除了在字符类规范中，正则表达式中的所有大写字符与小写相映射。可以使用 **noignorecase** (**noic**) 来撤消它。缺省值为 **noignorecase** 。

keyboardedit
设置此选项后，启用为了方便在命令模式下使用而在初始化时自动加载的编辑键映射。如果不设置此选项，将禁用（但不删除）映射。使用 **map** 命令可获得当前启用的命令模式映射列表。可以使用 **nokeyboardedit** 来撤消它。缺省值为 **keyboardedit** 。

keyboardedit!
设置此选项后，启用为了方便在命令模式下使用而在初始化时自动加载的编辑键映射。如果不设置此选项，将禁用（但不删除）映射。使用 **map!** 命令可获得当前启用的命令模式映射列表。可以使用 **nokeyboardedit!** 来撤消它。对于键盘编辑键发送 HP 形式转义序列 (一个 ESC 后接单个字符）的终端，其缺省值是 **nokeyboardedit!** 。对于其他终端，缺省值是 **keyboardedit!** 。

lisp
为 **lisp** 源代码修改 **autoindent** 模式和可视模式中的 **(** 、 **)** 、 **[[** 、 **]]** 、 **{** 和 **}** 。可以使用 **nolisp** 来撤消它。缺省值为 **nolisp** 。

list
用 **^I** 表示的制表符显示所有已输出的行，行的末尾用 **$** 标记。可以使用 **nolist** 来撤消它。缺省值为 **nolist** 。

magic
作用于正则表达式和备用替换字符串中的字符解释（请参阅前面的“正则表达式”和“替换字符串”）。可以使用 **nomagic** 来撤消它。 **ex** 、 **vi** 和 **view** ，缺省为 **magic** 。 **edit** 和 **vedit** ，缺省为 **nomagic** 。

mesg
允许其他用户使用 **write** 命令（请参见 *write*(1) ）向终端发送消息，这可能会中断屏幕显示。如果取消这项设置 (**nomesg**) ，那么当用户在使用编辑器，将阻止其他系统用户在用户的终端上的写入权限。可以使用 **nomesg** 来撤消它。缺省值是 **mesg** 。

modelines (**ml**)
如果设置了此选项，那么当编辑器读取文件时，将在处理 **.exrc** 和 **EXINIT** 命令之后但在将编辑控制交给用户之前，执行嵌入在文件前五行和最后五行中的任何 **ex** 命令。在单个行中， **ex** 命令必须以 **ex:** 或 **vi:** 为前缀，而且必须以 **:** 结尾。内嵌命令的前面或后面可以是任何数量的除冒号 (**:**) 以外的其他字符。可以使用 **nomodelines** (**noml**) 来撤消它。缺省值是 **nomodelines** 。

novice
使用供初学者使用的编辑器版本，也称为 **edit** 或 **vedit** 。可以使用 **nonovice** 来撤消它。 **ex** 、 **vi** 和 **view** ,缺省为 **nonovice** 。 **edit** 和 **vedit** ，缺省为 **novice** 。

number (**nu**)
输出行和行号。可以使用 **nonumber** (**nonu**) 来撤消它。缺省值是 **nonumber** 。

optimize (**opt**)
禁止在不支持直接光标定位的终端上使用自动回车。在某些情况下，如输出包含起始空白字符的多行时，将提高文本的输出效率。可以使用 **nooptimize** (**noopt**) 来撤消它。缺省值是 **nooptimize** 。

paragraphs=*pair-string* (**para**)
本选项的值是一个字符串这个字符串的连续字符对指定了文本处理宏的名称。（宏是 **.***xx* 形式的文本， **.** 是行的第一个字符）。

如果宏有一个单字符名称，使用空格替换名称中缺失的第二个字符。若要在这种替换中使用空格字符，前面必须加反斜杠 (\)，以防止编辑器将它解释为分隔符。

缺省值是 **paragraphs=IPLPPPQPP\ LIpplpipnpbp** 。

prompt
设置此选项后，用冒号 (**:**) 提示命令模式输入，如果不设置此选项，则不显示提示。可以用 **noprompt** 来撤消它。缺省值是 **prompt** 。

readonly (**ro**)
为正在编辑的文件设置 **readonly** 标志，从而防止在会话结束时发生意外覆盖。这个选项与使用 **-R** 调用 **ex** 、 **edit** 、 **vi** 和 **vedit** 是等同的，它还等同于使用 **view** 命令。可以使用 **noreadonly** (**noro**)
来撤消它。 **ex** 、 **edit** 、 **vi** 和 **vedit** ，缺省为 **noreadonly** 。 **view** 缺省为 **readonly** 。

redraw
在简易型终端上模拟智能终端。在输入模式下，输入文本时持续重复输出行。由于这可能需要在终端上进行大量的输出，因此只有在高传输速率下，它才是有用的。如果设置了 **noredraw** ，只有在输入模式终止后，而且在左边空白处用 **@** 标记删除行后，才重新输出行。可以用 **noredraw** 来撤消它。缺省值是 **redraw** 。

remap
如果设置了此选项，允许对使用其他宏定义的宏进行宏转换；转换将一直进行，直到获得最终结果。如果没有设置此选项，只进行一次转换。可以使用 **noremap** 来撤消它。缺省值是 **remap** 。

report=*n*
n 的值指定为了在行上显示报告而必须先使用命令来修改的行的数量。如果 *n* 是 5，那么就报告要修改 6 行或更多行。 **ex** 、 **vi** 和 **view** ，缺省为 **report=5** 。 **edit** 和 **vedit** ，缺省为 **report=1** 。

scroll=*n*
n 的值确定 **^D** 命令滚动的行数和 **z** 命令显示的行数（是滚动行数的两倍）。缺省值是 **window** 选项的值的一半。

sections=*pair-string*
此选项的值是一个字符串，在这个字符串中，连续的字符对指定启动各个节的文本处理宏的命称。请参阅前面的 **paragraphs** 编辑器选项。缺省值为 **sections=NHSHH\ HUuhsh+c**。

shell=*filename* (**sh**)
设置 **!** Shell 转义和 **shell** 命令使用的 Shell 名称。其缺省值是用户的 **SHELL** 环境变量，如果没有设置此选项，缺省值是 **/usr/bin/sh** 。

shiftwidth=*n* (**sw**)
设置 **autoindent** 和转移 (**<** 和 **>**) 命令所使用的缩进步长值。缺省值是 **shiftwidth=8** 。

showmatch (**sm**)
在可视模式下，如果屏幕上有相匹配部分，就短暂地跳转至相匹配部分 **(** 或 **{** （如果用户输入一个 **)** 或 **}** ）。可以使用 **noshowmatch** (**nosm**) 来撤消它。缺省值是 **noshowmatch** 。

showmode (**smd**)
在可视模式和开放模式下，在屏幕的右下角显示当前编辑器模式（如 **INPUT MODE** 、 **REPLACE 1 CHAR** 和 **REPLACE MODE** ）。可以使用 **noshowmode** (**nosmd**) 来撤消

它。 **ex** 、 **vi** 和 **view** ，缺省为 **noshowmode** 。 **edit** 和 **vedit** ，缺省为 **showmode** 。

slowopen （**slow**） 在可视模式下， **slowopen** 防止屏幕在输入过程不断刷新，以提升非智能终端的处理能力。可以使用 **noslowopen** （**noslow**) 来撤消它。缺省值是 **noslowopen** 。

tabstop=*n* （**ts**） 设置为编辑器在输入文件中扩展制表符使用的软件制表位的间隔。缺省值是 **tabstop=8** 。

taglength=*n* （**tl**） 设置标记中有意义字符的最大数量。多余的字符将被忽略。如果值为零，表示标记中的所有字符都是有意义的。缺省值是 **taglength=0** 。

tags=[*filename*]... 指定 **tag** 命令和 **-t** 命令行选项使用的标记文件。缺省值是 **tags=tags /usr/lib/tags** ，指定当前目录下的 **tags** 文件和 **/usr/lib/tags** 文件。文件名之前用空格分隔。

标记文件的每一行都包含以下三个用空格分隔的字段：标记名、要编辑的文件的名称和地址规范（请参阅前面的“地址”）。 **tags** 文件必须按标记名称排序。

ctags 命令（请参阅 *ctags*(1) ）用 C、PASCAL 和 FORTRAN 源文件创建标记文件。

tagstack （**tgst**） 启用被激活标记的下堆栈。可以使用 **notagstack** （**notgst**) 来撤消它。缺省值是 **tagstack** 。

当用户输入一个行模式 **tag** 命令或可视模式 **^]** 命令时，当前行号和文件名被存储在标记栈上。后面输入的行模式 **pop** 命令或可视模式 **^T** 命令将返回至已存储行号处的已存储文件名。

如果标记栈被禁用，那么将再次启用它，栈也恢复到上一次操作后的状态。如果禁用了标记栈， **pop** 命令将失效。

term=*termtype* 定义编辑器正在使用的终端类型。缺省值是从 **TERM** 环境变量中获得的。如果 **TERM** 没有设置或者为空， **term** 设置为 **unknown** 。 **term** 与 **ttytype** 编辑器选项之间没有差别。设置其中任何一个选项，将导致这两个选项的设置都被改变。

terse 使用短出错消息。可以使用 **noterse** 来撤消它。缺省值是 **noterse** 。

timeout （**to**） 如果设置了此选项，则要求在 **timeoutlen** 选项指定的时间内 接收一个多字符宏名称（ **map** 命令中的第一个参数），才能与宏名称进行匹配。如果不设置此选项，将不限制应在多长时间内完成宏名称的接收。可以使用 **notimeout** (noto) 来撤消它。缺省值是 **timeout**。

timeoutlen=*n* 以毫秒为单位设置宏的超时期限（请参阅 **timeout** 编辑器选项）。除非设置了 **timeout** ，否则此选项无效。 *n* 的值不能小于 1。缺省值是 **timeoutlen=500** （半毫秒）。

ttytype=*termtype* （**tty**） 定义编辑使用的终端的类型。有关详细信息，请参阅 **term** 编辑器选项。 **term** 与 **ttytype** 编辑器选项之间没有差别。设置其中任何一个选项，将导致这两个选项的设置都被改变。

warn 如果在被加载或完全写入之前文件，工作区发生了变化，那么在执行 **!** 或 **shell** 命令转义之前，将显示 **[No write since last change]** 消息。可以使用 **nowarn** 来撤消它。缺省值是 **warn** 。

window=*lines*（**wi**） 设置可视模式下文本窗口的行数。缺省值是不低于终端屏幕大小的值（定义在 **LINES** 环境变量（如果设置了此变量）中，否则缺省值是终端上 *terminfo*(4) 数据库中的条目）。但是，如果终端的波特率（请参阅 *stty*(1) ）低于 1200 或 2400，缺省值相应地减少为 8 行和 16 行。可以使用 **-w** 命令行选项设置启始值。

w300=*lines* 如果终端的波特率小于 1200，将 **window** 编辑器选项设置为指定的值。

w1200=*lines* 如果终端的波特率大于或等于 1200 但是小于 2400，将 **window** 编辑器选项设置为指定的值。

w9600=*lines* 如果终端的波特率大于或等于 2400，将 **window** 编辑器选项设置为指定的值。

wrapmargin=*n*（**wm**） 在可视模式下，如果 *n* 大于零，将在输入行的词界 (word boundary) 处自动插入换行符，从而使行与终端窗口的右边界相隔 *n* 个空格。缺省值是 **wrapmargin=0** 。

wrapscan（**ws**） 设置此选项后，使用 **/***re***/** （或 **?***re***?** ）的编辑器搜索从文件开头（或末尾）向文件的末尾（或开头）静默进行（不生成输出结果，也就是说，扫描被“包装”了）。当不设置此选项时，编辑器搜索相应地在文件的开头和末尾结束。可以使用 **nowrapscan**（**nows**) 来撤消它。缺省值是 **wrapscan** 。

writeany（**wa**） 禁止在写入命令前进行检查，从而允许对任何文件进行写入（在系统允许的情况下）。可以使用 **nowriteany**（**nowa**) 来撤消它。缺省值是 **nowriteany** 。

外部语言环境影响

环境变量

COLUMNS 此变量覆盖系统选择的水平屏幕大小。

LINES 覆盖系统选择的水平屏幕大小，用作可视模式下全屏或垂直屏幕的行数。

PATH 为 Shell 命令确定在编辑器选项、 **shell** 、 **read** 和 **write** 中指定的搜索路径。

SHELL 是一个变量，使用 **!** 、 **shell** 、 **read** 和带有 **!string** 形式操作数的其他命令时，应将它作为解释为首选的命令行解释器。对于 **shell** 命令，应使用单个参数 **-i** 来调用命令解释程序。对于其他命令，应使用 **-c** 和字符串来调用。如果没有设置 **SHELL** 环境变量，那么它被设置空字串，系统将使用 **sh** 实用程序。

TERM 是一个变量，它应被解释为终端的类型名称。如果这个变量没有设置或者为空，将使用缺省和终端类型。

EXINIT 是一个变量，应解释为一个 **ex** 命令列表，这些命令在编辑器启动后读取第一个文件之前执行。通过使用竖线字符 (|) 进行分隔，列表可以包含多个命令。

HOME 应被解释为目录名称，应在这个目录搜索编辑器启动文件名 **.exrc** 。

LC_COLLATE 用于确定正则表达式计算和 **tags** 文件处理中使用的排序序列。如果该选项没有指定或者为空，缺省值是 **LANG** 的值。

LC_CTYPE 确定将字符解释为单字节字符或多字节字符，区分字符的大小写，转换字符的大小写和正则表达式中与字符类规范相匹配的字符。此选项如果没有指定或者为空，缺省值是 **LANG** 。

LANG 用于确定显示消息的语言。此选项如果没有指定或者为空，缺省值是“C”（请参阅 *lang*(5) ）。

LC_ALL 确定用于覆盖各语言环境类别的任何值（通过设置 **LANG** 或以 **LC_** 开头的任何环境变量来指定）的语言环境。

LC_MESSAGES 确定确认响应的处理和使用何种语言编写消息。

如果任一国际化变量包含无效设置，则所有国际化变量将缺省为“C”（请参阅 *environ*(5)）。

设置此选项后， **TMPDIR** 环境变量指定临时文件所使用的目录，并覆盖缺省目录 **/var/tmp** 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

异步事件 （适用于 **XPG4**）

收到下列信号后将作出响应的动作：

SIGINT

发生中断时， **ex** 向终端报警，并发出消息。当前编辑器命令异常中止，而且 **ex** 退回命令行，并提示用户输入命令。如果标准输入不是终端设备，中断发生时 **ex** 将退出并返回非零退出状态。

SIGCONT

应刷新屏幕。

SIGHUP

如果上次 **e** 或 **w** 命令执行后，修改当前缓冲区， **ex** 将试图保存当前文件，以后可以用 **ex -r** 命令来恢复这个保存文件。

没有说明接收其他信号后应采取的行动。

扩展说明（仅适用于 **XPG4**）

ex 正在编辑的文件的路径名用 **current** 文件表示。文件的文本应读入到文件的工作版本（在这里称为 **buffer** ），所有的编辑修改都应在工作版本上进行，在使用 **ex** 命令将文件写出之前，所作的修改不影响原始文件。缓冲区中的行最多只能包含 4096 个字符，包括 2-3 个字节的开销。因此，如果行包含的字符超过 4092 个，将会产生问题。在编辑过程中，如果超出限制，将报告出错消息。

alternate 路径名是编辑器命令使用过的的前一文件的名称，如果使用过的最后一个文件成为当前文件，那么这个路径名就是前一当前路径名。当 **%** 出现在作为命令参数入的路径名之中时，将用 altername 路径名替换它。任何字符（包括 **%** 和 **#** ）的前面是反斜杠时，都将保留其字符值。

当发生错误时， **ex** 将向终端报警，并写出消息。

如果系统崩溃，而且修改没有被写入， **ex** 将试图保存缓冲区的内容。命令行选项 **-r** 可以用来提取被保存的修改。

初始化过程中（读取首个文件或处理终端的用户命令之前），如果设置了环境变量 **EXINIT** ，编辑器将执行包含在这个变量中的 **ex** 命令。如果没有设置这个变量， **ex** 将试图从 **$HOME/.exrc** 读取命令。只有当 **EXINIT** 或 **$HOME/.exrc** 设置了 exrc 编辑器选项时，ex 才试图读取当前目录下的 **.exrc** 文件中的命令。如果没有设置 **EXINIT** ，而且当前目录是用户的主目录，那么 **.exrc** 文件只能被处理一次。只有 **.exrc** 文件的所有者的用户 ID

与进程的有效用户 ID 相同，才能读取这个文件。处理 **.exrc** 文件后，应处理由 **-c** 选项指定的命令。

缺省情况下， **ex** 应在命令模式下启动，并用“:”提示符来标识。可以使用 **append** 、 **insert** 或 **change** 命令来进入输入模式。还有一种模式，即可视模式，在这种模式下，可以进行全屏幕编辑。 **visual** 命令对此进行了更为详细的介绍。命令行可以包含用竖线符 (|) 分隔的多个 **ex** 命令。除非它们是一行的最后命令，否则用这种方式进入输入模式或可视模式，将产生不可预见的结果。

应忽略以双引号 (**"**) 开头的命令。这可用作编辑器脚本中的注释。

警告

undo 命令将丢失被改变和恢复的行的所有标记。

z 命令输出多个逻辑行，而不是物理行。如果出现长行，将产生超过一整屏的输出。

空字符在输入文件中被忽略，而且不能出现在结果文件中。

在某些系统上，只有在系统启动时执行了某些与系统相关的动作，才能使用 **-r** 选项来恢复编辑文件。

编辑保留文件只有保存在运行相同 HP-UX 版本的系统上时，才能对它们进行恢复。不能在不同版本之间恢复保留文件。

在 HP 终端上，由 **map #***n* ... 命令指定的功能键的属性字段必须设置为 **normal** ，而不能设置为缺省的 **transmit**。

不要使用 **-C** 选项来编辑未加密的文件。 **-C** 选项只能用于已加密的文件。如果 **-C** 选项用于未加密的文件，编辑过程中的写入命令可能会破坏文件。

有关行长度限制和文件大小限制等方面的信息，请参阅警告一节中的 *vi*(1) 。

返回值（仅适用于 **XPG4**）

ex 实用程序用下列值中的一个值来退出：

0 成功完成。

>0 发生了错误。

作者

ex 由加州大学伯克利分校开发。 **ex** 的 16 位扩展部分以 Toshiba 公司的软件为基础。

文件

$HOME/.exrc	主编辑器的初始化文件
./.exrc	第二编辑器的初始化文件
/usr/lbin/expreserve	保存命令
/usr/lbin/exrecover	恢复命令
/usr/share/lib/terminfo/*/*	终端功能描述
/var/preserve	保存目录
/var/tmp/Ex*nnnnn*	编辑器临时文件

/var/tmp/Rx*nnnnn* 命名缓冲区临时文件

另请参阅

ctags(1)、 ed(1)、 stty(1)、 vi(1)、 write(1)、 terminfo(4)、 environ(5)、 lang(5)、 regexp(5)。

《The Ultimate Guide to the vi and ex Text Editors》， Benjamin/Cummings Publishing Company， Inc.， ISBN 0-8053-4460-8，HP 部件号 97005-90015。

符合的标准

ex：SVID2、SVID3、XPG2、XPG3、XPG4

名称

expand、unexpand - 将制表符扩展为空格以及相反操作

概要

expand [**-t** *tablist*] [*file* ...]

unexpand [**-a**] [**-t** *tablist*] [*file* ...]

过时形式:

expand [**-***tabstop*] [**-***tab1***,***tab2***,**...**,** *tabn*] [*file* ...]

说明

expand 可处理指定文件或标准输入，并将制表符更改为空格后写入标准输出。退格符会保留在输出中，在计算制表符时将列数减 1。为了正确计算制表符，如果一个多列字符将被“退格”，则它应该后跟与其列宽相等的多个退格符。如果在最后一个制表符位置后发现一个制表符，则将用单个空格替换它。对于在排序、查看特定列等之前预处理包含制表符的字符文件， **expand** 很有用。

expand 可识别下列命令行选项和参数:

[**-t** *tablist*] *tablist* 指定在何处设置制表符位置而不是使用缺省值 **8** 。 *tablist* 可以采用两种形式。如果它是单个数字，则按每隔 *tablist* 个空格设置制表符。 *tablist* 也可以是由空白或逗号分隔的列表，其中制表符的设置位置是递增的。

[**-***tabstop*] 此选项已过时，它等效于使用 **-t** *tabstop* 。

[**-***tab1***,***tab2***,**...**,***tabn*]
此选项已过时，它等效于使用 **-t** *tab1***,***tab2***,** ... **,***tabn* 。

unexpand 可处理指定的文件或标准输入，并尽可能将空格更改为制表符写入标准输出。缺省情况下，仅将前导空格和制表符转换为最大制表符字符串。缺省制表符位置是每 8 个字符。在输出中会保留退格符，在制表符计算中会将列数减 1。为了正确计算制表符，如果一个多列字符将被“退格”，则它应该后跟与其列宽相等的多个退格符。

unexpand 可识别下列命令行选项和参数:

-a 每当通过在制表符位置之前替换两个或多个空格而压缩生成的文件时，都会插入制表符。

-t *tablist* *tablist* 指定制表符位置。 *tablist* 可以采用两种形式。如果它是单个数字，则按每隔 *tablist* 个空格设置制表符。如果 *tablist* 是包含由空白或逗号分隔的递增位置的列表，则将在这些位置设置制表符。 **-t** 选项隐含 **-a** 选项。如果未指定 **-t** 选项，则缺省时等效于指定 **-t 8** （除了在此情况下不隐含 **-a** 外）。

外部语言环境影响

环境变量

LC_CTYPE 用于确定将文本解释为单字节字符和（或）多字节字符。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或 **LC_MESSAGES** ，或者将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值"C"（请参阅 *lang*(5)）而非 **LANG** 。

如果任一国际化变量包含无效设置，则 **expand** 和 **unexpand** 就会认为所有国际化变量都设置为"C"。请参阅 *environ*(5) 。

如果将 **LC_ALL** 设置为非空字符串值，则它将覆盖所有其他国际化变量的值。

国际代码集支持

支持单字节字符代码集和多字节字符代码集，但 **unexpand** 不识别多字节替代空格字符。

符合的标准

expand：XPG4、POSIX.2

unexpand：XPG4、POSIX.2

名称

expand_alias - 递归扩展 sendmail 别名

概要

expand_alias [**-r***max_recursion*] [**-t**] [**-tt**] *alias*

说明

Expand_alias 是一个 Shell 脚本，可递归扩展 **sendmail** 别名。使用 **telnet host 25** 及 **expn** 命令可使每个别名递归地扩展到其一个或多个目标中。每一级递归以缩进格式显示。由于递归地使用了 **telnet** ，因此 **expand_alias** 的速度很慢。如果由于防火墙的配置使本地 **telnet** 无法直接连接到远程系统，则 **expand_alias** 将不会成功执行。如果本地 **telnet** 将跨越防火墙透明地连接到远程系统，则 **expand_alias** 将可以与防火墙外的 sendmail 守护程序通信，使别名得以更完全地扩展（例如，某些本地 telnet 客户端使用位于防火墙上的 **socksd** 来允许本地 **telnet** 客户端透明地连接到 Internet 主机。如果本地缺省的 **telnet** 以这种方式使用 **socksd** ，则 **expand_alias** 会使用该 **telnet** 功能更完全地扩展别名）。

max_recursion 的缺省值为 10。在扩展了 *max_recursion* 指定的次数后，将不会再继续进行扩展。

如果指定了 **-t** ，将仅显示终端别名。

-tt 的作用与 **-t** 相似，区别在于如果终端行使用了管道，则会禁止输出该终端行，而是输出其上一级扩展。

举例

expand_alias root

expand_alias root@cat

expand_alias root@cat.cup.hp.com

expand_alias root@cup.hp.com

作者

expand_alias 由 HP 开发。

名称

expr - 将参数视为表达式进行计算

概要

expr *arguments*

说明

expr 将 *arguments* 视为表达式、对其进行计算，然后将结果写入到标准输出。表达式中的各元素必须以空白分隔。 Shell 特殊字符必须被转义。请注意， **0** 而不是空字符串被返回以表示零值。包含空白或其他特殊字符的字符串应该用单引号括起来。整型参数前可以有一个一元减号。在系统内部，整数被视为 2 的 32 位补数。

下面列出了运算符和关键字。需要转义的字符将以 \ 为前缀。该列表按优先级递增的顺序排列的，优先级相同的运算符则用 **{}** 符号组成组。

expr **\|** *expr* 如果第一个 *expr* 既不为空也不是 **0**， 则返回它。否则返回第二个 *expr* 。

expr **\&** *expr* 如果任一 *expr* 为空或为 **0**， 就返回第一个 *expr* ， 否则返回 **0** 。

expr { **=**, **\>**, **\>=**, **\<**, **\<=**, **!=** } *expr*

如果所有参数都是整数，且成功完成了比较，则 *expr* 返回 **1** ， 否则返回 **0** 。如果两个参数都是整型， *expr* 返回整型比较的结果，否则返回词汇比较结果（注意 **=** 和 **==** 是相同的，都用于比较是否相等）。

expr { **+**, **-** } *expr* 十进制整型值参数的加法与减法。

expr { *****, **/**, **%** } *expr*

结果为整型的十进制整型值参数的乘法、除法或求余运算。

expr **:** *expr* 匹配运算符 **:** 将第一个参数与第二个参数做比较，第二个参数必须是正则表达式。 *expr* 支持基本正则表达式的语法（请参阅 *regexp*(5)）， 除非所有模式都是"锚定"的（也就是说，以 ^ 开头）， 因此 ^ 在该上下文中不是特殊字符。通常，匹配运算符会返回匹配的字符数量（若失败，则返回 0）。另外，可以使用 **\(** ... **\)** 模式符号来返回第一个参数的一部分。

length *expr* *expr* 的长度。

substr *expr expr expr*

获取第一个 *expr* 的子字符串，这个字符串应该是以第二个 *expr* 指定的特殊字符开头，长度由第三个 *expr* 给定。

index *expr expr* 返回在第二个 *expr* 中找到的字符在第一个 *expr* 中的位置。

match Match 是一种前缀运算符，等效于中缀运算符 **:** 。

\(... **\)** 分组符号。任何表达式都可以放在括号中。括号可以嵌套，深度最大可达到头文件 **<limits.h>** 中指定的 **EXPR_NEST_MAX** 值。

外部语言环境影响

环境变量

LC_COLLATE 确定评估正则表达式时的排序序列，以及关系运算符在比较字符串值时的行为。

LC_CTYPE 用于确定将文本解释为单字节和（或）多字节字符，并且字符由正则表达式中的字符类别表达式匹配。

LANG 用于确定显示消息的语言。

如果未在环境中指定 **LC_COLLATE** 或 **LC_CTYPE** ，或将其设置为空字符串时， **LANG** 的值可作为各未指定或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **expr** 就会认为所有国际化变量都设置为“C”（请参阅 *environ*(5) ）。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

作为表达式计算的副作用， *expr* 会返回如下退出值：

0 表达式非空非零。

1 表达式为空或为零。

2 无效表达式。

>2 计算表达式时出错。

诊断信息

syntax error 运算符或操作数出错

non-numeric argument 试图对字符串进行算术运算

举例

将 shell 变量 **a** 加 1：

a=‘expr $a + 1‘

对于等于 **/usr/abc/file** 或 **file** 的 **$a** 来说，应返回路径名的最后一段（例如， **file** ）。请注意， **/** 在单独使用时是参数，因为 *expr* 将它解释为除法运算符（请参阅下面的 警告）。

expr $a : '.*/\(.*\)' \| $a

前例的一种更好的形式。 **//** 字符的增加消除了除法运算符的歧义，并且简化了整个表达式：

expr //$a : '.*/\(.*\)'

返回 **$VAR** 中的字符数：

expr $VAR : '.*'

警告

Shell 处理参数后， *expr* 只能通过值来判断运算符和操作数之间的不同。如果 **$a** 是一个 **=** ，则命令：

expr $a = '='

类似于：

expr = = =

当将参数传递给 *expr* （它们都将作为 **=** 运算符）时。将运行以下命令：

expr X$a = X=

作者

expr 由 OSF 和 HP 开发。

另请参阅

sh(1)、 test(1)、 environ(5)、 lang(5)、 regexp(5)。

符合的标准

expr：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

factor、primes - 对一个数进行因子分解，得到所有质数

概要

factor [*number*]

primes [*start* [*stop*]]

说明

如果命令行上没有提供任何参数，则 **factor** 会等待键入一个数。如果键入了正数，则会对该数进行因子分解并输出质因子；每个质因子将输出适当的次数。然后它会等待输入另一个数。如果 **factor** 遇到零或任何非数字字符，则会退出。

如果命令行上提供了参数，则 **factor** 会如上所述对该数进行因子分解，然后退出。

因子分解所需的最大时间与 sqrt(*n*) 成比例，并当 *n* 为质数或质数的平方时会达到最大时间。

factor 可处理的最大数为 1.0e14。

primes 输出上限和下限之间的质数。如果命令行上没有提供任何参数，则 **primes** 会等待键入两个数。第一个数被解释为下限；第二个数解释为上限。生成的范围（包括边界值）中的所有质数都将输出。

如果指定了 *start* 值，则会输出所有大于或等于 *start* 值的质数。如果指定了 *start* 值和 *stop* 值，则输出从 *start* 至 *stop* 范围（包括边界值）中出现的所有质数。

start 和 *stop* 值必须是以长整数表示的整数。

如果在上述任一情况中省略了结束值，则 **primes** 会一直运行，直到发生溢出或被键入的中断字符终止。

primes 可处理的最大数为 2147483647。

诊断信息

当输入值超出范围或遇到了非法字符，或者 *start* 值大于 *stop* 值时，这两个命令都会输出 **Ouch** 。

举例

输出数字 12 的质数因子分解:

factor 12

输出 0 和 20 之间的所有质数:

primes 0 20

名称

fastbind - 准备不完整的可执行文件以加快程序启动

概要

fastbind [-nu] *incomplete-executable*...

说明

fastbind 工具可以通过将有关所需共享库符号的信息存储在可执行文件来加快使用共享库的程序（不完整的可执行文件）的启动速度。

fastbind 将分析用于绑定可执行文件及其所有相关共享库的符号，并将该信息存储在可执行文件中。下次运行可执行文件时，动态加载程序将注意到该信息可用，并将使用该 fastbind 信息绑定可执行文件，而不是使用标准的搜索方法来绑定符号。对于基于 Itanium 的系统，动态加载程序是 **/usr/lib/hpux32/dld.so** （32 位模式）或 **/usr/lib/hpux64/dld.so** （64 位模式）。对于 PA-RISC 系统，动态加载程序是 **/usr/lib/dld.sl** （32 位模式）或 **/usr/lib/pa20_64/dld.sl** （64 位模式）。

由于 **fastbind** 将 fastbind 信息写入可执行文件，您必须对该可执行文件具有写权限。例外，如果所分析的可执行文件作为另一个进程运行，则将根据内核所做的修改锁定该文件，而 **fastbind** 将失败。

如果在创建 fastbind 信息后修改可执行文件所依赖的共享库，动态加载程序将以静默方式恢复到标准的搜索方式来绑定符号。 fastbind 信息可以通过对可执行文件- 再次运行 **fastbind** 来重新创建。 **fastbind** 将自动清除旧的 fastbind 信息并生成新信息。

ld 选项 **+fb** 可用于指示链接程序对它生成的不完整可执行文件运行 fastbind 工具。

环境变量

如果 **dld** 确定 fastbind 信息已过时，它将以静默方式恢复到标准的搜索方法来绑定符号。如果环境变量 **_HP_DLDOPTS** 设置为 **-fbverbose** ，动态加载程序将在 fastbind 信息过时的时候发出警告消息。

环境变量 **_HP_DLDOPTS** 可以设置为 **-nofastbind** ，使动态加载程序忽略 fastbind 程序并恢复到标准的搜索方法来绑定符号。

选项

fastbind 可识别下列选项：

-n 从可执行文件中删除 fastbind 信息，将其返回到最初对其运行 **fastbind** 之前所处的相同状态。

-u 通常，如果 **fastbind** 在绑定 fastbind 信息时检测到任何条件未满足的符号，则将生成错误消息，而不修改可执行文件。但是当 **fastbind** 通过 **-u** 选项调用时，则允许未解析的符号。

外部语言环境影响

环境变量

下列国际化变量会影响 **fastbind** 的执行：

LANG 当未提供 **LC_ALL** 和其他 **LC_*** 环境变量时，确定本国语言、本地惯例和编码字符集的语言环境类别。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省的 **C** （请参阅 *lang*(5) ）而非 **LANG**。

LC_ALL

确定所有语言环境类别的值，其优先级高于 **LANG** 和其他 **LC_*** 环境变量。

LC_MESSAGES

确定语言环境，用于影响写入标准错误中的诊断消息的格式与内容。

LC_NUMERIC

确定数字格式化时所用的语言环境类别。

LC_CTYPE

确定字符处理功能的语言环境类别。

NLSPATH

为处理 **LC_MESSAGES** 确定消息目录的位置。

如果任一国际化变量包含无效设置，则 **fastbind** 就会认为所有国际化变量均设置为 **C** 。请参阅 *environ*(5) 。

此外，下列环境变量也会影响 **fastbind** ：

TMPDIR

为临时文件指定一个目录（请参阅 *tmpnam*(3S) ）。

诊断信息

fastbind 在操作成功时返回零。非零的返回代码指示已出错。

举例

要对可执行文件 **a.out** 运行 **fastbind** ，请输入：

fastbind a.out

要在随后从可执行文件 **a.out** 中删除 fastbind 信息，请输入：

fastbind -n a.out

警告

32 位 PA-RISC **fastbind** 不能与 EXEC_MAGIC 可执行文件一起使用。

fastbind 有效地强制受限绑定和立即绑定。例如，考虑延迟的可执行文件链接绑定，它调用在隐式加载库中定义的函数 foo()。在进行实际的调用之前，如果显式地加载包含 **foo()** 定义的共享库（使用 *shl_load*(3X) 和 **BIND_FIRST** ），则在最终调用 **foo**() 时，将从显式加载的库中对其进行解析。但在运行 **fastbind** 之后，符号 **foo**() 将从隐式加载的库中进行解析。

作者

fastbind 由 HP 开发。

文件

a.out	输出文件
/usr/lib/dld.sl	32 位 PA-RISC 动态加载程序

/usr/lib/hpux32/dld.so	基于 32 位 Itanium 的动态加载程序
/usr/lib/pa20_64/dld.sl	64 位 PA-RISC 动态加载程序
/usr/lib/hpux64/dld.so	基于 64 位 Itanium 的动态加载程序
/usr/lib/nls/$LANG/fastbind.cat	消息清单
/var/tmp/__FB*	临时文件

另请参阅

系统工具

ld(1) 调用链接编辑器

其他信息

a.out(4) 汇编程序、编译程序和链接程序输出

dld.sl(5) PA-RISC 动态加载程序

dld.so(5) 基于 Itanium 的动态加载程序

文本和教程

《HP-UX Linker and Libraries User’s Guide》

名称

fastmail - 快速批处理邮件接口

概要

fastmail [**-b** *bcc-list*] [**-c** *cc-list*] [**-C** *comments*] [**-f** *from-name*] [**-F** *from-addr*] [**-i** *in-reply-to*] [**-r** *reply-to*] [**-R** *references*] [**-s** *subject*] *filename address-list*

说明

fastmail 命令是邮件系统的简单接口，该接口可以用来发送邮件，而不需要交互式邮件程序的开销。尤其是针对很多组人的批处理邮件非常有效。

所有的地址均应是完整的电子邮件地址、 **sendmail** 别名（位于 **/etc/mail/aliases** 文件中）或本地登录名。

选项

fastmail 识别下列选项:

-b *bcc-list* 包括一个 **Bcc:** 标题条目。将邮件副本发送到 *bcc-list* 中用逗号分隔的一系列地址。

-c *cc-list* 包括一个 **Cc:** 标题条目。将邮件副本发送到 *cc-list* 中用逗号分隔的一系列地址。

-C *comments* 包括一个 **Comments:** 标题条目（其中具有字符串值 *comments* ）。

-d 调试。显示关于处理步骤的信息。

-f *from-name* 将 **From:** 标题条目中的用户姓名替换为 *from-name* 。

如果用户为 **x@y** 且用户姓名为 **MrX** ，则缺省的 **From:** 行如下:

From: x@y (MrX)

-f Joe 选项将其更改为:

From: x@y (Joe)

-F from-addr 将 **From:** 标题条目中的地址替换为 *from-addr* 。在以上 **-f** 示例中， **-F a@b** 将源条目更改为

From: a@b (MrX)

-i *in-reply-to* 包括 **In-Reply-To:** 标题条目（其中具有字符串值 *in-reply-to* ）。这通常用来标识要回复的邮件。

-r *replyto* 包括 **Reply-To:** 标题条目（其中具有 *replyto* 中给定的单个地址）。这是通常要发送回复的地址，而不是 **From:** 标题条目中给定的收件人地址，这在邮件列表中很常用。

-R *references* 包括 **References:** 标题条目（其中包含字符串值 *references* ）。

-s *subject* 包括 **Subject:** 标题条目（其中包含 *subject* 值）。如果忽略此选项，则发送的邮件中不包括主题条目。

操作数

fastmail 可识别下列操作数：

address-list **To:** 标题行的一个或多个以空格分隔的地址列表。这些是邮件的主要收件人。

filename 包含邮件的文件的名称，或从标准输入中读取的短划线 (**-**)。

举例

完全指定命令

此命令已指定每个选项。

```
fastmail \
  -b "bcc1,bcc2,bcc3,bcc4" \
  -C "Just a Comment" \
  -c "cc1,cc2,cc3,cc4" \
  -d \
  -F me@anotherhost.com \
  -f My Name \
  -i "Your recent message" \
  -R REF:13579 \
  -r oscar \
  -s "Testing fastmail" \
  message-file \
  addr1 addr2 addr3 addr4
```

联机执行将显示下列调试消息：

```
Mailing to addr1,addr2,addr3,addr4 cc1,cc2,cc3,cc4 bcc1,bcc2,bcc
3,bcc4 [via sendmail]
cat /tmp/fastmail.5578 message-file | /usr/sbin/sendmail addr1,a
ddr2,addr3,addr4 cc1,cc2,cc3,cc4 bcc1,bcc2,bcc3,bcc4
```

接收的邮件具有下列相关的标题条目：

```
From realsender@mycomputer.myhost.com Tue Oct 22 21:14:04 EDT 1996
Subject: Testing fastmail
From: me@anotherhost.com (My Name)
Reply-To: oscar@mycomputer.myhost.com
To: addr1@mycomputer.myhost.com, addr2@mycomputer.myhost.com,
    addr3@mycomputer.myhost.com, addr4@mycomputer.myhost.com
Cc: cc1@mycomputer.myhost.com, cc2@mycomputer.myhost.com,
    cc3@mycomputer.myhost.com, cc4@mycomputer.myhost.com
References: REF:13579
In-Reply-To: Your recent message
```

Comments: Just a Comment

未传输 **Bcc:** 标题条目。

批处理

假设您是 **big** 用户（位于 **big-machine** 计算机上），并且具有名为 **batch-mail** 的 Shell 脚本，其中包含下列行：

```
#
# Batch Mail - batch mailing of a file to a LOT of users
#
# Usage: batch-mail "<from>" "<subject>" <filename>

sender_copy=$LOGIN
replyto=The-Mr-Big-list

fastmail -b $sender_copy -r $replyto -f "$1" -s "$2" $3 person1
sleep 10
fastmail -r $replyto -f "$1" -s "$2" $3 person2
sleep 10
fastmail -r $replyto -f "$1" -s "$2" $3 person3
sleep 10
fastmail -r $replyto -f "$1" -s "$2" $3 person4
```

命令：

```
batch-mail "Mr. Big" "Warning to all" warning.text
```

将 **warning.text** 文件的副本邮寄到 **person1**、 **person2**、 **person3** 和 **person4**，以 10 秒钟为间隔。

$LOGIN 还在邮件中静默接收第一个邮件的副本。得到的每个邮件中均包含标题行：

```
From: big@big-machine (Mr. Big)
Subject: Warning to all
Reply-To: The-Mr-Big-list
```

文件

/etc/mail/aliases	**sendmail** 别名文件。
/usr/sbin/sendmail	邮件传输代理。
/tmp/fastmail.*pid*	临时文件。

作者

fastmail 由 HP 开发。

另请参阅

elm(1)、 sendmail(1M)。

RFC 822 “Internet 文本消息格式标准”

名称

file - 确定文件类型

概要

file [**-m** *mfile*] [**-c**] [**-f** *ffile*] [**-h**] *file* ...

说明

file 在尝试对每个 *file* 进行分类时先执行一系列测试。如果 *file* 是 ASCII 文件，则 **file** 会检查前 512 字节并尝试猜测其语言。如果 *file* 是可执行 **a.out** 文件，只要其版本标记大于 0， **file** 就会输出版本标记（请参阅 *ld*(1) 中 **-V** 的说明）。

file 使用文件 **/etc/magic** 标识包含某种 "幻数"的文件，即，包含可指示文件类型的数字或字符串常量的任何文件。 **/etc/magic** 的开头注释说明了其格式。

选项

file 可识别下列命令行选项：

-m *mfile* 使用备用 magic 文件 *mfile* 。

-c 检查 magic 文件有无格式错误。出于效率的考虑，通常不会执行此验证。指定此选项后，将不进行文件分类。

-f *ffile* 从文件 *ffile* 中获取要检查的文件列表。 **file** 会对 *ffile* 中出现其名称的每个文件进行分类。

-h 不处理符号链接。

外部语言环境影响

环境变量

LC_MESSAGES 用于确定显示的消息所用的语言。

如果在环境中未指定 **LC_MESSAGES** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省的"C"（请参阅 *lang*(5) ），而不使用 **LANG** 。

如果任一国际化变量中包含无效设置，则 **file** 就会认为所有国际化变量都设置为"C"。请参阅 *environ*5 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。但所有非 ASCII 文本文件将被标识为"数据"。

警告

某个版本的 **file** 命令可正确解释该特定版本的核心文件。对不同版本上生成的核心文件使用 **file** 命令将报告错误结果。

另请参阅

ld(1)。

符合的标准

file：SVID2、SVID3、XPG2、XPG4

名称

find - 查找文件

概要

find *pathname_list* [*expression*]

说明

find 命令以递归方式降序检索 *pathname_list* （即，一个或多个路径名）中给出的每个路径名的目录层次结构，以搜索与下述基元编写的布尔型 *expression* 相匹配的文件。缺省情况下， **find** 不使用符号链接。

布尔型表达式使用短路求值进行求值。这意味着，只要通过求值左侧参数可以求得布尔运算（AND 或 OR）的结果，则不再对右侧参数求值。

在基元的说明中，参数 *n* 表示十进制整数； **+***n* 表示大于 *n* ， **-***n* 表示小于 *n* ， *n* 就表示 *n* 。

该命令可以识别下列基元：

-depth 位置无关项，可使目录层次结构降序排列，以便在检索目录自身之前，先检索目录中的所有条目。当 **find** 与 *cpio*(1) 一起使用以传输包含在无写入权限的目录中的文件时，此选项十分有用。当使用 *cpio*(1) 同时必须保留目录的修改日期时，此选项也十分有用。始终为真。

-follow 位置无关项，可使 **find** 使用符号链接。使用符号链接时， **find** 会跟踪所访问的目录，从而可以检测到无限循环；例如，如果符号链接指向祖先，则将发生这样的循环。此表达式不应与 **-type l** 表达式一起使用。始终为真。

-fsonly *FStype* 位置无关项，当任一目录的文件系统不是 *FStype* 指定的文件系统时，使 **find** 停止对该目录进行降序排序，其中， *FStype* 是 **cdfs** 、 **hfs** 、 **vxfs** 或 **nfs** 之一，它们分别表示 CDFS、HFS、JFS (VXFS) 或 NFS 文件系统类型。

这种环境中，挂接点继承其父目录的 *FStype* 。这意味着，当 **-fsonly hfs** 已指定且 **find** 遇到挂接在 HFS 文件系统上的 NFS 挂接点时，将访问该挂接点，但不访问该挂接点下面的条目。请务必注意，当指定了 **-fsonly nfs** 时，NFS 文件系统的挂接点下面的任何 HFS 文件系统都不会被遍历。始终为真。

-local 当文件实际驻留在本地系统上时为真。此选项并没有限制只能搜索实际驻留在本地系统上的文件，它只不过与这类文件相匹配。请参阅举例。

-xdev 位置无关项，可使 **find** 避免跨越任何文件系统挂接点，这些挂接点位于 *pathname_list* 中枚举的起始点下面。将访问挂接点自身，但不访问该挂接点下面的条目。始终为真。

-mountstop 与 **-xdev** 相同。此基元仅用于向后兼容模式。 **-xdev** 的优先级高于 **-mountstop** 。

-name *file* 当模式 *file* 与当前文件名的最后一个组成部分相匹配时，此选项为真。根据文件名扩展的模式匹配表示法，对模式进行匹配。当从 Shell 中调用 **find** 时，模式应该进行转义（使用反斜杠）或使用引号括起来，以防止 Shell 扩展任何元字符。模式可能包含补充代码集字符。

-path *file* 与 **-name** 相同，只不过使用完整路径（将作为 **-print** 的输出）而不只是基名，请注意， **/** 字符不视为特例。例如， ***/.profile** 与 **./home/fred/.profile** 相匹配。

-perm [**-**]*mode* 在此基元中，参数 *mode* 用于表示文件模式位。除第一个字符不能是 **-** 运算符之外，该参数的格式与 *chmod*(1) 中所述的 *mode* 操作数的格式相同。当使用 *mode* 的符号格式时，假定起始模板清除了所有文件模式位。

如果省略前导负号，则当文件权限位与 *mode* 的值完全匹配时，此基元为真。当省略负号时，将忽略与符号属性 **s** （设置用户 ID、设置组 ID）和 **t** （粘着位）相关的位。

如果 *mode* 前面有负号，则当 *mode* 中设置的所有位也在文件权限位中进行了设置时，此基元为真。这种情况下，与符号属性 **s** 和 **t** 相关的位有意义。

-fstype *FStype* 当文件所属的文件系统是类型 *FStype* 时，此选项为真，其中， *FStype* 是 **cdfs** 、 **hfs** 、 **nfs** 或 **vxfs** 之一，它们分别对应于 CDFS、HFS、NFS 或 JFS (VXFS) 文件系统类型。

-type *c* 当文件类型是 *c* 时为真，其中 *c* 是下列值之一:

f	常规文件
d	目录
b	块设备专用文件
c	字符设备专用文件
p	FIFO（命名管道）
l	符号链接
s	套接字
n	网络设备专用文件
M	挂接点

-links *n* 当文件有 *n* 个链接时为真。

-user *uname* 当文件属于 *uname* 时为真。如果 *uname* 是数字，同时似乎不是 **/etc/passwd** 文件中的登录名，则该数字将被视为用户 ID。 *uname* 操作数之前可以加 **+** 或 **-** ，以修改这些基元的比较关系。如果参数 *n* 表示十进制整数；则 **+***n* 表示大于 *n* ， **-***n* 表示小于 *n* ， *n* 就表示 *n* 。

-group *gname* 当文件属于 *gname* 时为真。如果 *gname* 是数字，同时未出现在 **/etc/group** 文件中，则该数字将被视为组 ID。 *gname* 操作数之前可以加 **+** 或 **-** ，以修改这些基元的比较关系。如果参数 *n* 表示十进制整数；则 **+***n* 表示大于 *n* ， **-***n* 表示小于 *n* ， *n* 就表示 *n* 。

-nouser 当文件属于口令数据库中没有列出的用户 ID 时，此选项为真。请参阅 *passwd*(4) 。

-nogroup 当文件属于组数据库中没有列出的组 ID 时，此选项为真。请参阅 *group*(4) 。

-size *n*[**c**] 当文件长度为 *n* 块（每块为 512 字节）时，此选项为真。如果 *n* 后跟 **c** ，则文件大小用字节表示。

-atime *n* 当初始化时间减去文件访问时间后是 *24* 小时的 *n*-1 至 *n* 倍时为真。初始化时间是介于以下两个操作之间的时间： **find** 实用程序的调用与 **find** 实用程序的调用首次访问由其 **path** 操作数指定的任何文件。 *pathname_list* 中目录的访问时间可用 **find** 本身进行更改。

-mtime *n* 当初始化时间减去文件修改时间后是 *24* 小时的 *n*-1 至 *n* 倍时为真。初始化时间是介于以下两个操作之间的时间： **find** 实用程序的调用与 **find** 实用程序的调用首次访问由其 **path** 操作数指定的任何文件。

-ctime *n* 当初始化时间减去对文件状态信息进行最后一次更改所花的时间后，其结果是 *24* 小时的 *n*-1 至 *n* 倍时为真。初始化时间是介于以下两个操作之间的时间： **find** 实用程序的调用与 **find** 实用程序的调用首次访问由其 **path** 操作数指定的任何文件。

-newer *file* 当最近修改过的当前文件新于参数 *file* 时，此选项为真。

-newer[*tv1*[*tv2*]] *file* 若当前文件的指示时间值 (*tv1*) 新于 *file* 的指示时间值 (*tv2*) ，则此选项为真。时间值 *tv1* 和 *tv2* 均可从下列字符集中选择：

- **a** 文件的上次访问时间
- **c** 文件中 i 节点的上次修改时间
- **m** 文件的上次修改时间

如果省略 *tv2* 字符，则其缺省值为 **m** 。请注意， **-newer** 选项等效于 **-newermm** 。

语法示例；

-newera *file*
-newermc *file*

-inum *n* 当文件序号（i 节点编号）为 *n* 时，此选项为真。请注意，文件序号仅在给定文件系统内是唯一的。因此，文件序号相匹配并不保证所引用的文件是相同的，除非将搜索限定到单个文件系统。

-linkedto *path* 当文件与 *path* （即，链接至 *path* ）指定的文件是同一物理文件时，此选项为真。该基元与 **-inum** 类似，但是当文件硬链接至 *path* 时，它可以正确地检测，即使在搜索多个文件系统时，也是如此。

-print 可输出当前路径名。始终为真。

-exec *cmd* 当执行 *cmd* 后返回零值作为退出状态时为真。 *cmd* 的结尾必须使用分号 (**;**) 或加号 (**+**) （分号和加号是 Shell 专用的，必须进行转义）。使用 **+** 时， *cmd* 会聚集一组路径名并在其上执行。第一次出现的 **{}** 与 **+** 之间的任何命令参数将被忽略。 **+** 比 **;** 更受欢迎的原因是，它可大幅提高性能。所有命令参数 **{}** 将替换为当前路径名。 *cmd* 可能包含补充代码集字符。

-ok *cmd* 与 **-exec** 相同，但例外的是，生成的命令行首先输出一个问号，并且仅当用户键入 **y** 进行响应时才会执行。确定响应的格式与语言环境有关：在 C 语言环境中为 y，请参阅

environ(5) 中的 **LANG** 。 *cmd* 的结尾必须使用分号 **(;)** （分号是 Shell 专用的，必须进行转义）。 *cmd* 可能包含补充代码集字符。

-cpio *device* 以 *cpio*(4) 格式（5120 字节记录）将当前文件写入 *device* 。使用 **-cpio** 隐含了 **-depth** 的功能。始终为真。

-ncpio 与 **-cpio** 相同，但会将 **-c** 选项添加到 **cpio** 。使用 **-ncpio** 隐含了 **-depth** 的功能。始终为真。

-prune 如果当前条目是一个目录，则会使 **find** 跳过该目录。要避免遍历特定目录，或在使用 **cpio -p** 时避免递归循环，此选项十分有用。但请注意，如果还给定了 **-depth** 选项，则 **-prune** 无效。有关详细信息，请参阅下文的 **-only** 说明以及举例一节。始终为真。

-only 这是 **-prune** 的正逻辑版。在遍历每个目录后将执行 **-prune** ，除非针对相应目录 **-only** 已成功求值。例如，下列三个命令是等效的：

find . -fsonly hfs -print
find . -print -fstype hfs -only
find . -print ! -fstype hfs -prune

但请注意，如果还给定了 **-depth** 选项，则 **-only** 无效。始终为真。

(*expression* **)** 当括号中的表达式为真时，此选项为真。需要使用空格。括号对 Shell 是专用的，必须进行转义，与 **\(** 和 **\)** 中一样。

可以使用下列运算符组合使用基元（按照优先级递减的顺序）：

! *expression* 逻辑 NOT 运算符。当 *expression* 非真时，此选项为真。

expression [**-a**] *expression* 逻辑 AND 运算符。当两个 *expression* 都为真时，此选项为真。

expression **-o** *expression* 逻辑 OR 运算符。当两个 *expression* 中有一个为真或同时为真时，此选项为真。

如果忽略 *expression* ，或者 **-print** 、 **-ok** 、 **-exec** 、 **-cpio** 或 **-ncpio** 均未指定，则假定为 **-print** 。 **-user** 、 **-group** 和 **-newer** 基元会对各自的参数求值一次。

HFS 访问控制列表

-acl 基元使用户可以搜索 HFS 访问控制列表条目。如果文件的访问控制列表与访问控制列表模式相匹配，或者包含可选的访问控制列表条目（请参阅 *acl*(5) ），则此选项为真。它有三种格式：

-acl *aclpatt* 与其访问控制列表包含由 *aclpatt* 模式指定的所有（零个或多个）模式条目的所有文件相匹配。

-acl =*aclpatt* 仅当某个文件的访问控制列表包含由 *aclpatt* 模式指定的所有（零个或多个）模式条目，同时访问控制列表中的每个条目至少与 *aclpatt* 模式中指定的一个模式条目相匹配时，才与该文件相匹配。

-acl opt 与包含可选访问控制列表条目的所有文件相匹配。

可以将 *aclpatt* 字符串作为运算符或短格式模式给出；请参阅 *acl*(5) 。

缺省情况下，对于其访问控制列表包含 *aclpatt* 中所有（零个或多个）访问控制列表模式的文件， **-acl** 为真。文件的访问控制列表也可以包含不匹配的条目。

如果 *aclpatt* 以 **=** 开头，则该字符串的剩余部分必须与文件的访问控制列表中的所有条目相匹配。

aclpatt 字符串（缺省情况下，或开头有 **=** 时）可以是访问控制列表或访问控制列表模式。但是，如果它是访问控制列表，则 *aclpatt* 必须至少包含三个基本条目（ (*user*.%, 模式) 、 (%.*group*, 模式) 和 (%.%, 模式)） 。

特殊情况下，如果 *aclpatt* 是单词 **opt** ，则对包含访问控制列表条目的文件，该基元为真。

JFS 访问控制列表

-aclv 基元使用户可以搜索 JFS 访问控制列表条目。如果文件的访问控制列表与访问控制列表模式相匹配，或者包含可选的访问控制列表条目（请参阅 *aclv*(5) ），则此选项为真。它有三种格式：

-aclv *aclpatt* 与其访问控制列表包含由 *aclpatt* 模式指定的所有（零个或多个）模式条目的所有文件相匹配。

-aclv =*aclpatt* 仅当某个文件的访问控制列表包含由 *aclpatt* 模式指定的所有（零个或多个）模式条目，同时访问控制列表中的每个条目至少与 *aclpatt* 模式中指定的一个模式条目相匹配时，才与该文件相匹配。

-aclv opt 与包含可选访问控制列表条目的所有文件相匹配。

缺省情况下，对于其访问控制列表包含 *aclpatt* 中所有（零个或多个）访问控制列表模式的文件， **-aclv** 为真。文件的访问控制列表也可以包含不匹配的条目。

如果 *aclpatt* 以 **=** 开头，则该字符串的剩余部分必须与文件的访问控制列表中的所有条目相匹配。

aclpatt 由一个 *type* 字段、一个 *ID* 字段和一个 *mode* 字段组成，并用冒号分隔。可以指定用多个逗号分隔的 *aclpatt* 。

type 字段可以是 **user** 、 **group** 、 **class** 、 **other** 或 ***** 之一，可以选择以 **default:** 开头。 **user** 、 **group** 、 **class** 、 **other** 和 **default** 可以分别缩写成 **u** 、 **g** 、 **c** 、 **o** 和 **d** 。 ***** *type* 字段可以与上述任一类型相匹配。

ID 字段可以是数字形式的用户或组 ID、分别来自 **/etc/passwd** 或 **/etc/group** 的用户或组 ID 字符串，或是可以与任何 ID 相匹配的 ***** 。

mode 字段由三个字符的字符串组成。第一个字符可以是 **r** （表示授予读取权限）、 **-** （表示拒绝读取权限）或 **?** （与两种读取权限状态之一相匹配）。第二个字符可以是 **w** 、 **-** 或 **?** ，类似地表示写入权限状态；第三个字符可以是 **x** 、 **-** 或 **?** ，类似地表示执行权限状态。

特殊情况下，如果 *aclpatt* 是单词 **opt** ，则对包含可选访问控制列表条目的文件，该基元为真。

外部语言环境影响

环境变量

如果未指定国际化变量，或者其值为空，则缺省为 **LANG** 的值。

如果未指定 **LANG** ，或者其值为空，则缺省为 **C** （请参阅 *lang*(5) ）。

如果将 **LC_ALL** 设置为非空字符串值，则它将覆盖所有其他国际化变量的值。

如果任一国际化变量包含无效设置，则所有国际化变量将缺省为 **C** （请参阅 *environ*(5) ）。

LC_CTYPE 将文本解释确定为单字节和（或）多字节字符，将字符分类确定为可输出的，并确定符合正则表达式中字符类表达式的字符。

LC_MESSAGES 确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息以及写入到标准输出的信息消息的格式和内容。

NLSPATH 确定消息目录的位置，以便处理 **LC_MESSAGES**。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

搜索两个目录 **/example** 和 **/new/example** ，以查找包含字符串 **Where are you** 的文件，并输出这些文件的名称：

```
find /example /new/example -exec grep -l 'Where are you' {} \;
```

删除一周内没有访问过的文件名为 **a.out** 或 ***.o** 的所有文件：

```
find / \( -name a.out -o -name '*.o' \) -atime +7 -exec rm {} \;
```

请注意，需要使用空格以定界已转义的圆括号。

以单个长列表的形式输出当前目录中名为 ***.o** 的所有文件：

```
find . -name '*.o' -exec ls -l {} \+
find . -name '*.o' -exec ls -l \+
```

请注意，加号前面的大括号是可选的。

输出该计算机上所有文件的文件名。避免遍历 **nfs** 目录，同时仍输出 **nfs** 挂接点：

```
find / -fsonly hfs -print
```

仅匹配本地文件，不检查远程挂接的任何目录中的内容：

```
find / ! -local -prune -o -size +50 -print
```

仅当在远程目录的前端没有挂接本地文件系统时，它才能正确运行。本示例将输出系统上大于 50 块的所有本地文件，而不会浪费时间访问远程文件。

要取得相同效果，但检查远程目录上挂接的本地文件系统中的文件，请使用：

find / -local -size +50 -print

将整个文件系统复制到 **/Disk** 上安装的磁盘，同时避免递归复制问题。两个命令是等效的（注意，使用 **-path** 而不是 **-name** ）：

cd /; find . ! -path ./Disk -only -print | cpio -pdxm /Disk

cd /; find . -path ./Disk -prune -o -print | cpio -pdxm /Disk

将根磁盘复制到 **/Disk** 上安装的磁盘，并跳过 **/** 下面挂接的所有文件系统。请注意， **-xdev** 不会使 **/** 被跳过，即使它是一个挂接点也是如此。这是因为 **/** 是起始点，而 **-xdev** 仅影响起始点 下面的的条目。

cd /; find . -xdev -print | cpio -pdm /Disk

将目录子树中所有常规文件的权限更改为模式 444，将所有目录的权限更改为 555：

find *pathname* **-type f -print | xargs chmod 444**

find *pathname* **-type d -print | xargs chmod 555**

请注意， **find** 的输出是通过管道传送给 *xargs*(1) ，而不是使用 **-exec** 基元。这是因为，当单个命令要处理大量的文件或目录时， **-exec** 基元会为每个文件或目录产生单独的进程，但 *xargs* 可以将文件名或目录名收集到单个 *chmod* 命令的多个参数中，从而减少了进程数并提高了系统效率。对于 **-exec** 基元，可以使用 **+** 定界符获得相同的效率。

访问控制列表示例

查找不属于用户 **karl** 的所有文件，这些文件的访问控制列表中至少有一个条目与 **karl** 相关，同时有一个条目用于组 **bin** 中的非特定用户，并且其读取位打开、写入位关闭：

find / ! -user karl -acl 'karl.*, %.bin+r-w' -print

查找在任何访问控制列表条目中设置了读取位的所有文件：

find / -acl '*.*+r' -print

查找在每个访问控制列表条目中未设置写入位但已设置执行位的所有文件：

find / -acl '=*.*-w+x' -print

查找包含可选访问控制列表条目的所有文件：

find / -acl opt -print

相关内容

NFS

对于 NFS 文件， **-acl** 基元始终为假。

警告

为了实现互操作性， **cpio** 不支持大于 2GB 的归档文件或其用户（或）组 ID 大于 60000 (60K) 的文件。对于用户（或）组 ID 大于 60K 的文件，将以当前进程的用户（或）组 ID 进行归档和恢复。

作者

find 由 AT&T 和 HP 开发。

文件

/etc/group	组名
/etc/mnttab	挂接点
/etc/passwd	用户名

另请参阅

chacl(1)、 chmod(1)、 cpio(1)、 setacl(1)、 sh(1)、 test(1)、 xargs(1)、 mknod(2)、 stat(2)、 cpio(4)、 group(4)、 passwd(4)、 acl(5)、 aclv(5)、 environ(5)、 lang(5)、 regexp(5)。

符合的标准

find： SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

findmsg、dumpmsg - 为修改创建消息清单文件

概要

findmsg [**-aiv**] [[**-D** *sym*] [**-U** *sym*]] *file* ...

dumpmsg *file* ...

说明

findmsg 命令可从 C 程序源 *file* 提取消息，并将这些消息以适合作为 **gencat** （请参阅 *gencat*(1) ）输入的格式写入到标准输出。将使用 **cpp** （请参阅 *cpp*(1) ）对输入文件进行预处理，以便选择打印说明符和处理 **ifdef** 、 **ifndef**... 条件性 **cpp** 基元。如果指定了多个输入文件，且未使用 **-a** 选项，则将顺序处理这些文件，以便在输出每个输入 *file* 之前写入标识输入 *file* 的消息清单注释行。

findmsg 命令可扫描源文件，以查找未注释行，其中嵌有以下三种格式之一：

catgets(*any_var***,NL_SETN,***n***,<***message***>)**

<*message***>**   **/* catgets** *n* ***/**

/* catgets *n* ***/**   **<***message***>**

或嵌有完全包含在单个物理行上的这些格式的任意组合。 **<***message***>** 可能是一个字符串常量或字符串常量和打印说明符 (PRI*) 的组合。任意数量的空格或制表符都可以将 **catgets** 注释与 *message* 分隔开。数字 *n* （它可以是任何有效消息编号，请参阅 *gencat*(1) ）将与 *message* 字符串一起生成消息清单源行。消息源行将被指定给其编号为 **NL_SETN** 的当前值的集，如遇到的最后一个 **#define** 指令所设置的。如果在找到消息行时尚未定义 **NL_SETN**，则将输出消息，且不包含集编号规范。如果找到了属于同一集和消息编号的多条消息，则将输出找到的最后一条消息；以静默方式忽略任何其他消息。将忽略 C 源文件中的条件编译和 **#include** 指令。

选项

findmsg 可识别下列命令行选项：

-a   将多个输入文件中编号相同的集合并在一起，以便 **gencat** 可以处理 **findmsg** 输出。

-D*sym*   定义 *sym* 。

-U*sym*   导致 *sym* 是未定义的。

-i   考虑所有 #ifdefs 以便从输入文件中提取消息。如果未使用此选项，则将使用选项 **-D** 和 **-U** 来选择打印说明符。

-v   输出由 **cpp** 发出的所有错误消息。缺省情况下， **findmsg** 不显示由 **cpp** 发出的错误消息。

dumpmsg 命令可从 **gencat** 创建的消息清单 *file* 中提取消息。该命令将消息以适合于编辑和重新输入到 **gencat** 的格式写入标准输出。

外部语言环境影响

环境变量

LC_CTYPE 用于确定将文本解释为单字节字符和（或）多字节字符。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或 **LC_MESSAGES** ，或者将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值 **C** （请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **findmsg** 和 **dumpmsg** 将认为所有国际化变量都设置为 **C**。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

警告

findmsg 和 **dumpmsg** 命令是 HP 专有的，无法移植到其他供应商的系统，在将来的 HP-UX 版本中不再提供该命令。

作者

findmsg 和 **dumpmsg** 由 HP 开发。

另请参阅

findstr(1)、 gencat(1)、 insertmsg(1)、 catgets(3C)。

名称

findstr - 查找包括在消息清单中的字符串

概要

findstr *file* ...

说明

findstr 在 C 源代码中文件查找未注释的字符串常量，在标准输出中输出这些常量及其两端的引号，并在其前面输出文件名、起始位置和长度。 **insertmsg** 将使用该信息（请参阅 *insertmsg*(1) ）。 **findstr** 不输出 **catgets()** 例行程序参数字符串（请参阅 *catgets*(3C) ）。

外部语言环境影响

环境变量

LC_CTYPE 用于确定将注释和字符串文字解释为单字节字符和（或）多字节字符。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或 **LC_MESSAGES** ，或者将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省的“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量中包含无效设置，则 **findstr** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*（5）。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

警告

findstr
输出静态字符串变量的初始化字符串。使用这些字符串调用 **insertmsg** 将导致以 **catgets()** 调用替换它们（请参阅 *catgets*(3C) ）。因为初始设置必须为字符串，所以这种赋值将导致无效的 C 声明。例如，下一行：

static char *x[] = "*message***"**

会被 **insertmsg** （请参阅 *insertmsg*(1) ）修改为：

static char *x[] = (catgets(catd,NL_SETN,1,"*message***"))**

在这些字符串输入至 **insertmsg** 之前，应手动将其从 **findstr** 输出中删除。

未来的 HP-UX 版本将不提供 **findstr** 。

另请参阅

insertmsg(1)。

名称

finger - 用户信息查找程序

概要

finger [*options*] *user_name* ...

说明

缺省情况下， **finger** 列出系统中每个 *user_name* 的信息：

- 登录名，
- 完整用户名，
- 终端写入状态（如果写权限被拒绝），
- 空闲时间，
- 登录时间，
- 用户的主目录和登录 Shell，
- 用户在其主目录下的 **.plan** 文件中放入的任何计划，
- 用户参与的项目，来自主目录下的 **.project** 文件，
- 办公室地点和电话号码（如果知道），
- 用户上次接收邮件的时间，以及用户上次阅读邮件的时间。

如果空闲时间作为一个整数列出，则表示分钟；如果其中包含 **:** ，则表示小时与分钟；如果包含 **d** ，则表示天数和小时。可以接受用户的姓与名以及帐户名。

finger 还可用于列出远程计算机上的用户。 *user_name* 的格式为 *user_name@host* 。如果未指定 *user_name* ，则远程系统（HP-UX 或非 将使用缺省标准格式列出用户信息。

选项

finger 可识别下列选项：

-b 不输出用户的主目录和 Shell。

-f 不输出通常以短格式输出的标题。

-h 不以长格式输出 **.project** 文件。

-i 强制采用“空闲”输出格式。类似于短格式，但仅输出登录名、终端、登录时间和空闲时间。

-l 强制使用长输出格式。

-m 仅对用户名匹配参数。

-p 不输出 **.plan** 文件。

-q 强制使用快速输出格式。类似于短格式，但仅输出登录名、终端和登录时间。

-R 输出用户的主机名。

-s 强制使用短输出格式。

-w 不以短格式输出完整名称。

警告

只输出 **.project** 文件的第一行。

作者

finger 由加州大学伯克利分校开发。

文件

/etc/utmps	who 文件
/var/adm/wtmps	上次登录文件
/etc/passwd	用于用户名、办公室...
˜/.plan	计划
˜/.project	项目
/var/mail	邮件目录

另请参阅

chfn(1)、 who(1)、 utmpd(1M)。

名称

fmt - 格式化文本

概要

fmt [**-cs**] [**-w** *width*] [*file*...]

说明

fmt 命令是一种简单的文本格式化程序，它可以填充和连接各行来生成输出行，其最大长度为 **-w** *width* 选项所指定的字符数。缺省 *width* 是 72。 **fmt** 连接 **file** 参数。如果不指定任何选项， **fmt** 将格式化来自标准输入的文本。

输出中将保留空行，还保留单词之间的空隔。为了与 **nroff** 兼容， **fmt** 不会填充以句点 (**.**) 开头的行。该命令也不填充以 **From:** 开头的行。

在输出中将保留缩进，且不会连接具有不同缩进的输入行（除非使用了 **-c** ）。

fmt 也可用作 **vi** 的内置文本过滤器； **vi** 命令：

!}fmt

重新格式化介于光标位置和段落末尾之间的文本。

选项

fmt 可识别下列选项：

-c 指定段落缩进模式。保留一个段落内前两行的缩进格式，并使后面每个行的左边缘与第二行的左边缘对齐。该选项对于带有标记的段落很有用。

-s 仅拆分行。不将较短的行连接成较长的行。该选项可防止错误地连接示例代码行和其他具有这种“格式”的文本。

-w 填充输出行，直到其达到 *width* 所指定的宽度。

警告

目前为了与 BSD 兼容提供了 **-w** *width* 选项，但在以后的版本中可能不再提供该选项。

另请参阅

nroff(1)、 vi(1)。

名称

fold - 折叠较长的行，以便在有限宽度输出设备上显示

概要

fold [**-b**] [**-s**] [**-w** *width*] [*file* ...]

过时格式:

fold [**-s**] [**-***width*] [*file* ...]

说明

fold 命令是折叠指定文件中的内容的过滤器，可将行断开，使其最多具有 *width* 指定的列或字节数（如果指定了 **-b** 选项）。 **fold** 命令通过插入换行符断行，以便每个输出行具有可能的最大宽度（不超过指定的列数或字节数）。不能在字符中间断行。如果未指定文件或者指定的 *file* 名为 **-** ，则使用标准输入。

fold 命令通常用于将文本文件发送到行式打印机，此类打印机会截断而不是折叠比它能够打印的宽度更长的行。

如果在输入中遇到退格符、制表符或回车符，且未指定 **-b** 选项，则会对它们进行特殊处理，如下所示:

退格符　将行宽的当前计数减 1，但是该计数永远不会变为负数。因此，字符序列 回车符算作使用一个列，假定两个字符各占据一个列。 **fold** 不会紧接在任何退格符之前或之后插入换行符。

制表符　遇到的每个制表符会使列位置指针前移到下一个制表位。制表位设置为相隔 8 列，位于列位置 1、9、17、25、33 等。

回车符　行宽的当前计数设置为零。 **fold** 不会紧接在任何回车符之前或之后插入换行符。

请注意， **fold** 可能影响存在的任何下划线。

选项

fold 命令可识别下列选项和命令行参数:

-b　按字节而不是列位置计算 *width* 。

-s　在指定的列数（或字节数）之前找到的最后一个空白字符处断行。如果未找到，则按指定的行长度断行。

-w *width*
-*width*　按列位置或字节（如果指定了 **-b** ）指定最大行长度。缺省值是 80。如果存在制表符则 *width* 应该是 8 的倍数，或者在 **fold** 处理制表符之前应该使用 **expand** 来扩展制表符（请参阅 *expand*(1) ）。 **-***width* 选项已过时，可能会在将来的版本中删除。

外部语言环境影响

环境变量

LC_CTYPE 用于确定将文本解释为单字节字符和（或）多字节字符。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或 **LC_MESSAGES** ，或者将其设置为空字符串，则会将 **LANG** 的值用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅

lang(5) ）而非 **LANG** 。

如果任一国际化变量中包含无效设置，则 **fold** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

另请参阅

expand(1)。

符合的标准

fold：XPG4、POSIX.2

名称

forder - 转换文件数据顺序

概要

forder [**-a**] [**-l**] [**-n**] [*file* ...]

说明

文件的文本方向（模式）可以是从右到左（非拉丁模式）或从左到右（拉丁模式）。该文本方向可能会影响数据在文件中的排列方式。所得到的数据排列称作屏幕顺序和键盘顺序。 **forder** 将文件中字符的顺序从屏幕顺序转换为键盘顺序或者反向转换。

forder 读取输入文件的连接形式（如果未给定，则读取标准输入），然后在标准输出上生成其输入的转换版本。如果 **-** 用作输入文件名， **forder** 将在该点处读取标准输入（使用 **--** 可在这些情况下界定选项的结束）。

forder 转换所有从右到左读取的语言的输入文件。除非使用 **-a** 选项，否则该命令仅将输入文件复制到从左到右读取的语言的标准输出。

选项

forder 可识别下列选项：

- **-a** 转换所有从左到右读取的语言的文件数据顺序。
- **-l** 将文件标识为已在拉丁模式下创建。
- **-n** 将文件标识为已在非拉丁模式下创建。

外部语言环境影响

环境变量

LANGOPTS 环境变量确定文件的模式和顺序。 **LANGOPTS** 的语法为：

[*mode*] [*_order*]

其中的 *mode* 描述文件的模式： **l** 表示拉丁模式， **n** 表示非拉丁模式。对于 **l** 和 **n** 之外的值，假定为非拉丁模式。 *order* 描述文件的数据顺序： **k** 为键盘， **s** 为屏幕。对于 **k** 和 **s** 之外的值，假定为键盘顺序。 **LANGOPTS** 中的模式信息可以从命令行上覆盖。

LC_ALL 环境变量确定语言的方向（从左到右或从右到左）。

LC_NUMERIC 环境变量确定语言是否具有备用的数字系统。

LANG 环境变量确定显示消息所用的语言。

国际代码集支持

支持单字节字符代码集。

举例

以下命令以采用屏幕顺序的 *file1* 开始，将其转换为键盘顺序，将输出按键盘顺序排序，将其转换回屏幕顺序，然后将输出重定向到 *file2* 。请注意， **-n** 用于通知 **forder** *file1* 是在非拉丁模式下创建的。

forder -n file1 | sort | forder -n > file2

警告

用户应确保 **LANGOPTS** 环境变量准确反映文件的状态。

如果存在，备用的数字始终具有从左到右的方向。

forder 命令是 HP 专有程序，不可移植到其他供应商的系统上，并且将不在将来的 HP-UX 版本中提供。

作者

forder 由 HP 开发。

另请参阅

environ(5)、 strord(3C)、 nljust(1)。

名称

from - 谁是发件人？

概要

from [**-s** *sender*] [*user*]

说明

from 通过在邮箱文件中显示邮件标题行来显示邮件的发件人。如果指定了 *user* ，则将检查该 *user* 的邮箱而不检查您自己的邮箱。如果指定了 **-s** 选项，则只显示来自 *sender* 的邮件的标题。

举例

列出邮箱中所有由 **ken** 发送的当前邮件的标题行。

from -s ken

文件

/var/mail/*

作者

from 由加州大学伯克利分校开发。

另请参阅

biff(1)、 mail(1)、 prmail(1)。

名称

fruled - 闪烁（或关闭）警示指示灯（单元、机柜和 I/O 机箱警示指示灯）

概要

fruled -c *cell* [**-c**...] [**-f**|**-o**] [**-B**]
　　[**-u** *username*:[*passwd*] **-h** *IPaddress*|*hostname*
　　| **-g** [*passwd*] **-h** *IPaddress*|*hostname*]

fruled -i *I/Ochassis* [**-i**...] [**-f**|**-o**] [**-B**]
　　[**-u** *username*:[*passwd*] **-h** *IPaddress*|*hostname*
　　| **-g** [*passwd*] **-h** *IPaddress*|*hostname*]

fruled -b *cabinet* [**-b**...] [**-f**|**-o**]
　　[**-u** *username*:[*passwd*] **-h** *IPaddress*|*hostname*
　　| **-g** [*passwd*] **-h** *IPaddress*|*hostname*]

fruled -C|**-I** [**-l** *cabinet*] [**-l**...] [**-f**]
　　[**-u** *username*:[*passwd*] **-h** *IPaddress*|*hostname*
　　| **-g** [*passwd*] **-h** *IPaddress*|*hostname*]

说明

fruled 命令可使本地组合系统中单元或 I/O 机箱的警示指示灯闪烁（或关闭）。该命令也可用于使机柜号指示灯开始闪烁或停止闪烁。如果指定了 **-u** 选项或 **-g** 选项，则命令的范围将是指定的目标组合系统。

如果指定单元或 I/O 机箱警示指示灯闪烁，也可以使用 **-B** 选项使包含该单元或 I/O 机箱的机柜的机柜号指示灯闪烁。同样，如果关闭单元或 I/O 机箱的指示灯，也可以使用 **-B** 选项使机柜号指示灯停止闪烁。

注释： " 该 "命令不读取或显示任何指示灯的状态。用户必须观察硬件本身来确定其状态。

有关本联机帮助页中所用分区管理术语的说明，请参阅《HP 系统分区指南》。

选项和参数

fruled 识别下列命令行选项和参数：

-f　关闭指定的警示指示灯。这是缺省值。

-o　使指定的警示指示灯开始闪烁。

　-o 选项不能与 **-C** 或 **-I** 一起使用。 **-f** 和 **-o** 选项相互排斥。

-B　开始或停止包含单元或 I/O 机箱的机柜的机柜号指示灯的闪烁。 **-B** 选项只能与 **-c** 和 **-i** 选项一起使用。

-u *username*:[*passwd*]
　指定访问本地分区之外的分区所需的授权（但也可用作对本地分区的回送访问权）。要修改的组合系统是该目标分区驻留的组合系统。

如果使用该选项，则必须使用 **-h** 选项。

username 指定目标分区上的配置用户名。

passwd 指定与 *username* 相关联的口令。如果该字段为空，则命令将提示指定口令。

注释：该命令是基于 Web 的企业管理 (WBEM) 客户端应用程序。 **-u** 选项使用安全套接字层 (SSL) 连接来访问目标分区。如果报错，请检查是否满足相关内容部分所述的条件。

安全警告：在命令行上直接指定口令可能会在环境中造成安全风险。调用 *ps*(1) 或其他相关命令时，可以显示进程的命令行。这种情况下，系统上任何已验证的用户均可能在进程执行过程中看见口令。因此，强烈建议不要在命令行上指定口令，而应该让命令提示口令。

-h *IPaddress*|*hostname*

该选项只应与 **-u** 或 **-g** 选项组合使用。 *IPaddress*|*hostname* 指定目标分区 (**-u**) 或组合系统 (**-g**) 的 IP 地址或主机名。

-g [*passwd*] 允许访问由 **-h** 选项指定的组合系统。所访问的组合系统将被当作目标组合系统。访问是通过服务处理器的 LAN 端口进行的。

如果使用该选项，则必须使用 **-h** 选项。

passwd 指定服务处理器的 IPMI 口令。如果省略该字段，则命令将提示指定口令。

如果在尝试使用该选项进行连接时报错，请检查并确保远程服务处理器上没有禁用 IPMI LAN 访问。通过登录服务处理器并使用 Command Menu 中的 **SA** 命令，可以启用或禁用通过 LAN 上的 IPMI 对组合系统进行的访问。

-u 和 **-g** 选项相互排斥。

安全警告：在命令行上直接指定口令可能会在环境中造成安全风险。调用 *ps*(1) 或其他相关命令时，可以显示进程的命令行。这种情况下，系统上任何已验证的用户均可能在进程执行过程中看见口令。因此，强烈建议不要在命令行上指定口令，而应该让命令提示口令。

-c *cell* 闪烁（或关闭）指定的 *cell* 警示指示灯。

cell 可以按照本地 (cabinet#/slot#) 或全局 (cell#) 格式来指定。例如，位于机柜 0 插槽 4 的单元可以在本地标识为 0/4，在全局仅标识为 4。

-i *I/Ochassis* 闪烁（或关闭）指定的 *I/Ochassis* 警示指示灯。

I/Ochassis 必须按照 *cabinet#/enclosure#/chassis#* 格式来指定。例如，位于机柜 0 机壳 1 和 I/O 机箱插槽 1 的 I/O 机箱标识为 0/1/1。

-b *cabinet* 开始或停止指定 *cabinet* 的机柜号指示灯的闪烁。

-C 关闭所有单元警示指示灯。

-I 关闭所有 I/O 机箱指示灯。

-l *cabinet* 将 **-C** 或 **-I** 选项的范围限制到给定的 *cabinet* 。

全局单元号到本地单元号的对应关系

组合系统中的机柜从 0 开始编号。每个机柜中的单元插槽也从 0 开始编号。每个机柜最多可以包含 8 个单元。例如，位于机柜 0 中的单元将具有如下全局格式的单元号：0, 1, 2, 3, 4, 5, 6, 7. 相应本地格式的单元号将是 0/0、0/1、0/2、0/3、0/4、0/5、0/6、0/7。

同样，位于机柜 1 中的单元将具有如下全局格式的单元号：8, 9, 10, 11, 12, 13, 14, 15. 相应本地格式的单元号将是 1/0、1/1、1/2、1/3、1/4、1/5、1/6、1/7。

根据上述惯例，位于机柜 1 插槽 0 的单元在本地格式中标识为 1/0，在全局格式中标识为 8。 *parstatus*(1) 命令会将上述单元显示为“cab1,cell0”。位于机柜 1 插槽 4 的单元在本地格式中标识为 1/4，在全局格式中标识为 12。*parstatus*(1) 命令会将上述单元显示为“cab1,cell4”。

返回值

fruled 命令退出时返回下列值之一：

0 成功完成。

1 发生了错误。

2 没有指示灯与指定对象相关联。

举例

使位于机柜 0 插槽 4 的单元的警示指示灯闪烁，也使包含它的机柜的警示指示灯闪烁。

fruled -o -B -c 0/4

关闭位于机柜 0 插槽 4 和机柜 0 插槽 6 的两个单元的警示指示灯。

fruled -f -c 0/4 -c 0/6

相关内容

该命令使用 基于 *Web* 的企业管理 *(WBEM)* 产品及其某些配置设置。如果在使用 **-u** 选项时遇到错误，请检查是否满足以下两个条件：

- 使用 *cimconfig*(1M) 命令验证（在必要时更正）以下两个变量的的设置：
  - **enableRemotePrivilegedUserAccess=true**
  - **enableHttpsConnection=true**
- 必须已经将目标分区的数字证书追加到本地分区的 Trust Store 文件。对于 nPartition 命令， Trust Store 文件是 **/var/opt/wbem/client.pem** 。

  注释：必须已经将目标分区的数字证书追加到本地分区的 Trust Store 文件。对于 npartition 命令，Trust Store 文件是 **/var/opt/wbem/client.pem** 。该文件由 WBEM 安装附带的命令使用。因此，如果 WBEM 安装附带的命令信任某个目标分区，则 npartition 命令也将信任该目标分区。

有关其他信息，请参阅下文另请参阅部分中列出的 WBEM 文档。

作者

fruled 由 HP 开发。

另请参阅

frupower(1M)、 parcreate(1M)、 parmgr(1M)、 parmodify(1M)、 parremove(1M)、 parstatus(1)、 parunlock(1M)、 partition(5)、

docs.hp.com 上的《HP 系统分区指南》

docs.hp.com 上的《HP WBEM Services for HP-UX System Administrator's Guide》

docs.hp.com 上的《HP WBEM Services for HP-UX 11i v2.0 on Integrity Servers Version A.01.05 Release Notes》

名称

ftio - 快速磁带 I/O

概要

ftio -o|**-O** [**achpvxAELM**] [**-B** *blksize*] [**-D** *type*] [**-e** *extarg*] [**-K** *comment*] [**-L** *filelist*]
[**-N** *datefile*] [**-S** *script*] [**-T** *tty*] [**-Z** *nobufs*] *tapedev* [*pathnames*] [**-F** *ignorenames*]

ftio -i|**-I** [**cdfmptuvxAEMPR**] [**-B** *blksize*] [**-S** *script*] [**-T** *tty*] [**-Z** *nobufs*] *tapedev* [*patterns*]

ftio -g [**v**] *tapedev* [*patterns*]

说明

ftio 是一种专为将文件复制到磁带机而设计的工具。在相似情况下，它的执行速度比 **cpio** 或 **tar** 快（请参阅 *cpio*(1) 和 *tar*(1)）。 **ftio** 使用多个进程来读/写文件系统和写/读磁带设备，使用进程间共享的大量内存以及较大块来读写磁带。

ftio 与 **cpio** 兼容，因为 **cpio** 的输出始终可以由 **ftio** 读取，且 **ftio** 的输出可以由 **cpio** 读取，除了在此联机帮助页后面的“cpio 兼容性”一节中说明的情况以外。

必须使用下列选项之一调用 **ftio**： **-o** 、 **-O** 、 **-i** 、 **-I** 或 **-g** 。 **-o** 和 **-O** 选项指示 **ftio** 从文件系统输出文件到磁带； **-i** 和 **-I** 选项指示 **ftio** 从磁带读取文件到文件系统。 **-o** 、 **-O** 、 **-i** 和 **-I** 选项后可以跟修饰符，该修饰符必须紧接在选项之后，且选项和修饰符之间不能有空格，如 **ftio -idxE** （请参阅下面的“修饰符”一节）。

tapedev 指定要向其写入输出的磁带设备的设备专用文件名。可以按以下格式指定远程计算机上的设备：

*machine***:***device_special_file*

ftio 通过远程计算机上的 **/usr/sbin/rmt** 创建一个服务器进程以访问磁带设备。如果在远程系统上不存在 **/usr/sbin/rmt** ，则 **ftio** 将通过远程计算机上的 **/etc/rmt** 创建一个服务器进程以访问磁带设备。

选项

ftio 可识别下列选项：

-o 将文件（包括路径名和状态信息）从文件系统复制到 *tapedev* 。如果指定了 *pathnames*，则 **ftio** 以递归方式向下遍历 *pathnames* 来查找文件，并将这些文件复制到 *tapedev* 中。如果未指定 *pathnames* ，则 **ftio** 将读取标准输入以获取要复制的路径名的列表。如果需要， **ftio** 可以将文件复制到多个磁带中。对于使用的每个磁带， **ftio** 都会生成一个磁带头，其中包含当前的磁带卷号、计算机节点名和类型、操作系统名、发行版号和版本号（均来自 **uname()** 系统调用；请参阅 *uname*(2)）、发出 **ftio** 命令的用户的用户名、命令的执行时间和日期、已经连续使用当前介质的次数、一个注释字段和 **ftio** 在内部使用的其他项目。带头与磁带归档的主体由文件结束标记分隔。通过将设备文件名作为第一个参数调用 **cat** （请参阅 *cat*(1)），可以读取磁带头。请注意，使用 **-o** 选项写入的字符设备和块设备专用文件无法传输到其他 HP-UX 实现。

-O 当未对 **-O** 使用修饰符时，将以与 **ftio -ocva** 相同的方式将文件复制到磁带设备。但是，如果 **.ftiorc** 文件存在于用户的主目录中，则 **ftio** 会打开此文件，并进行扫描以查找前面

有 **O=** 的行。在匹配行上定义的选项将被传递给 **ftio** ，就像已在命令行上指定了这些选项。请参阅 举例一节。

-i 从 *tapedev* 提取文件（即复制到文件系统中），假定该设备为磁带和以前的 **ftio -o** 操作的产品。根据模式匹配表示法（请参阅 *regexp*(5) ）的规则，只会选择其文件名与 *patterns* 相匹配的文件。此外，模式中的前导 **!** 表示仅应该选择与模式的其余部分 不匹配的那些名称。可以指定多个 *patterns* 。如果未指定 *patterns* ，则 *patterns* 的缺省值为 ***** （即选择所有文件）。所提取的文件将有条件地创建并复制到当前目录树中，视下述选项而定。这些文件的权限是以前 **-o** 操作的权限。

-I 当未对 **-I** 使用修饰符时，将以与 **ftio -icdmv** 相同的方式提取文件（即复制到文件系统中）。但是，如果 **.ftiorc** 文件存在于用户的主目录中，则 **ftio** 会打开此文件，并进行扫描以查找前面有 **I=** 的行。在匹配行上定义的选项将被传递给 **ftio** ，就像已在命令行上指定了这些选项。请参阅 举例一节。

-g 读取 *tapedev* 中的文件列表。如果指定了 *patterns* ，则仅打印输出匹配的文件名。请注意，文件名前面始终有在创建文件列表时 **ftio** 期望文件所在的卷；因此在这一意义上只有最后一个卷是有效的。

-e *extarg* 指定如何处理要归档的文件的任何盘区属性。使用 **ftio.** 归档文件时，无法保留盘区属性。 *extarg* 采用下列值之一：

- **warn** 发出一条警告消息，并归档没有盘区属性的文件。
- **ignore** 将归档具有盘区属性的文件，但不保留盘区属性，且不发出警告消息。
- **force** 不归档具有盘区属性的文件，但发出警告消息。

如果未指定 **-e** ，则 *extarg* 的缺省值为 **warn** 。

-B *blksize* 指定写入磁带的块的大小（字节）。此数值可以以 **k** 结束，表示乘以 1024。使用较大的块通常可提高性能和磁带使用率。允许的最大块大小受所用的磁带机限制。设置的缺省值是 16 384 字节，因为这是大多数 HP 磁带机上的最大块大小。

-D *type* 仅当某一目录所属的文件系统为 *type* （其中 *type* 可以是 **hfs** 、 **vxfs** 或 **nfs** ）时，才以递归方式向下遍历该目录。

-F *ignorenames* 跟在 **-F** 后面的参数可指定不应该复制到磁带的 *patterns* 。适用于 *ignorenames* 的规则与适用于 *patterns* 的规则相同；请参阅前面有关 **ftio -i** 的说明。

-K *comment* 指定要放在 **ftio** 磁带头中的注释。

-L *filelist* 创建正在备份的文件的列表。 *filelist* 指定了输出文件。如果指定了 *pathnames* ，则在实际开始备份之前，执行文件搜索并生成一个文件列表。然后，此列表会被追加到备份中每个磁带的磁带头，作为 **ftio** 试图复制到此磁带上的文件的列表。备份过程中的最后一盘磁带包含用于标识文件在归档集中位置的目录。如果也没有指定 *pathnames* ，则在开始备份之前，从标准输入获取该文件列表。除了生成文件列表外， **-L** 选项还可实现磁

带检查点功能，允许在损坏的介质上写入失败时重新开始备份。

-M 使该命令与 **cpio** 完全兼容。也就是说，不生成或不期望磁带头，并将缺省块大小更改为 5120 字节。（请参阅下面的“cpio 兼容性”一节）。

-N *datefile* 仅将比 *datefile* 中指定的文件更新的文件复制到磁带。

-R 当 **ftio** 出现不同步时自动进行重新同步。从多磁带备份中除第一盘磁带外的其他磁带进行恢复时，该选项很有用。缺省情况下， **ftio** 将询问用户是否需要重新同步。

-S *script* 指定在多磁带备份中每完成一盘磁带时要调用的命令。该命令使用下列参数进行调用：*script tape_no user_name* 。 *script* 是 **-S** 选项指定的字符串参数 *script* 。 *tape_no* 是所需磁带的编号， *user_name* 是调用 **ftio** 的用户。通常，字符串 *script* 可指定用于通知用户需要更改磁带的 Shell 脚本。

-T *tty* 指定 **/dev/tty** 的替代项。 **ftio** 通常在需要进行终端交互时打开 **/dev/tty** 。

-Z *nobufs* 指定要用作两个进程间缓冲空间的 *blksize* 内存组块数，其中 *blksize* 是写入到磁带的块的大小。组块通常越多越好，但是达到某个点时性能不会再有任何改进，而且当从主内存中分出一部分作为缓冲区时性能可能会下降。为 *nobufs* 设置的缺省值是 16，但是，如果系统的负载不是很重，则使用 32 或 64 可能会提高性能。当系统处于单用户模式下执行备份时可获得最佳结果（请参阅 *shutdown*(1M) ）。

修饰符

下列修饰符可以与某些选项一起使用，如 概要中所示：

a 将文件复制到磁带后，重置其访问时间，使得这些文件似乎未曾被 **ftio** 访问。

c 以 ASCII 字符形式写入头信息，以便可以进行移植。

d 在恢复文件时，根据需要创建目录。

f 除了与 *patterns* 匹配的文件外，将所有其他文件复制到文件系统中。

h 将符号链接指向的文件视为普通文件或目录进行归档。缺省情况下， **ftio** 会归档链接本身。

m 保留以前的文件修改时间和文件所有权。恢复修改时间不适用于正在恢复的目录。

p 在备份即将结束时，打印输出已传输的块数、所用总时间（排除磁带倒带和换带时间），以及通过这些数字计算得到的有效传输速率。如果指定了两次 **p** ，则将在每盘磁带的末尾打印输出这些值。

t 仅打印输出输入的目录。不创建、读取或复制文件。

u 无条件复制（缺省情况下， **ftio** 不会用较旧的文件替换同名的新文件）。

v 使用详细模式。打印输出文件名和磁带头的列表。当与 **t** 修饰符一起使用时，目录看起来与 **ls -l** (ell) 命令的输出相同（请参阅 *ls*(1) ）。

x 保存或恢复设备专用文件。 **ftio** 使用 *mknod*(2) 在恢复操作过程中重新创建这些文件。因此，只有具有适当特权的用户能够使用此修饰符。该修饰符常用于系统内备份。将设备文件恢复到其他系统上可能是非常危险的。

A 如果从磁带上复制文件 （**-i** 或 **-I** 选项），则打印输出在磁带归档上找到的所有文件名，并指明哪些文件已经恢复。当用户恢复选定的文件，但是希望知道哪些文件（如果有的话）在磁带上时，这是很有用的。

如果将文件复制到磁带 （**-o** 或 **-O** 选项）上，则 **A** 修饰符会禁止输出有关可选访问控制列表条目的警告消息。 *ftio*(1) 不备份文件访问控制列表中的可选访问控制列表条目（请参阅 *acl*(5)）。通常，对于具有可选访问控制列表条目的每个文件，会打印输出警告消息。

E 在归档时，使用相对于根目录的路径名来存储具有绝对路径名（以 **/** 开头的路径名）的所有文件，即删除前导的 **/** 。在恢复时，会将归档之前在归档中具有绝对路径名的任何文件恢复为具有相对于当前目录的路径名。

L 与 **-L** 选项基本相同，不同之处是，文件列表作为文件 **ftio.list** 而不是 *filelist* 中指定的文件保留在当前目录中。

P 在恢复时，使用 **prealloc()** 预先为文件分配磁盘空间（请参阅 *prealloc*(2)）。这会大大改进文件段的本地化。

在到达磁带末尾时，如果指定了 **-S** 选项， **ftio** 会调用 *script* 并将当前磁带倒带，然后要求用户安装下一盘磁带。

要将一个或多个元字符传递给 **ftio** 而不让 Shell 扩展它们，请通过在其中每个元字符前面添加反斜杠（例如 **/usr*** ）或者将其用保护性单引号括起（例如 **'/usr*'** ）来保护它们。

cpio 兼容性

ftio 使用与 **cpio** 相同的归档格式。但是， **ftio** 在缺省情况下创建磁带头并使用大小为 16 KB 的磁带块。而 **cpio** 在缺省情况下使用 512 字节的块。当与 **-B** 选项一起使用时， **cpio** 使用 5120 字节的块。要在输入或输出模式下实现与 **cpio** 的完全兼容，用户应该指定 **M** 修饰符。 **ftio -oM** 会创建不具有磁带头的单磁带或多磁带归档，并且在缺省情况下与 **cpio -[o|i]B** 创建相同的块大小。可以使用 **ftio -iM** 来恢复 **cpio -oB** 命令创建的归档。如果组合使用 **ftio** 的 **M** 修饰符与 **-B 512** 块大小规格，则可实现与 **cpio -[o|i]** （无 **-B** ）的完全兼容。

外部语言环境影响

环境变量

LC_COLLATE 可确定为生成文件名评估模式匹配表示法时使用的排序序列。

LC_CTYPE 可确定与模式匹配表示法中的字符类表达式匹配的字符。

LC_TIME 可确定日期和时间字符串的格式和内容。

LANG 用于确定显示消息的语言。

如果未在环境中指定 **LC_COLLATE** 、 **LC_CTYPE**
或 **LC_TIME** ，或者将其设置为空字符串，则会将 **LANG** 的值用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则使用缺省值 C（请参阅 *lang*(5)）而非 **LANG** 。如果任一国际化变量中

包含无效设置，则 **ftio** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集。

举例

将文件系统的全部内容（包括专用文件）复制到磁带机 **/dev/rmt/c0t0d0BEST** 上：

ftio -ox /dev/rmt/c0t0d0BEST /

恢复 **/dev/rmt/c0t0d0BEST** 上的所有文件，相对于当前目录：

ftio -idxE /dev/rmt/c0t0d0BEST

列出使用 **ftio -o** 创建的备份集的内容。请注意，使用 **v** 修饰符可以提供更详细的列表，并显示磁带头的内容。

ftio -itv /dev/rmt/c0t0d0BEST

显示如何使用 **.ftiorc** 文件：

假定 **.ftiorc** 文件存在于用户主目录中且包含以下内容：

```
# Sample .ftiorc file.
I= cdmuvEpp -B 16k -S /usr/local/bin/ftio.change
O= cavEpp -Z 8 -B 16k -S /usr/local/bin/ftio.change
```

使用以下命令行调用 **ftio** 来备份用户的主目录和操作系统命令目录：

ftio -O /dev/rmt/c0t0d0BEST /home/my_home /usr/sbin

指定 **-O** 选项会使 **ftio** 检查 **.ftiorc** 文件以查找其他选项。在这种情况下，将生成字符头、重置访问时间、将复制的文件的列表输出到标准输出、将所有文件名（路径名相对于 **/** ）复制到 **/dev/rmt/c0t0d0BEST** ，并在备份完成时（以及每次更换磁带时）打印输出性能数据，如果备份时需要多个介质，则在完成每个介质后 **ftio** 将调用脚本 **/usr/local/bin/ftio.change** 。

警告

为了遵循行业标准和实现互操作性， **ftio** 不支持归档大于 2 GB 的文件或其用户（或组）ID 数大于 60 K 的文件。用户（或组）ID 数大于 60K 的文件将以当前进程的用户（或组）ID 进行归档和恢复。

ftio 使用 System V 共享内存和信号量运行。在进程终止时，系统不会自动释放提交给这些功能的资源。 **ftio** 仅在正常终止或收到以下信号之一后终止时才这样做： **SIGHUP** 、 **SIGINT** 、 **SIGTERM** 。任何其他信号都按 *signal*(2) 描述的缺省方式进行处理。请注意， **SIGKILL** 的行为是立即终止进程。因此，如果 **ftio** 收到 **SIGKILL** 信号（可能是由于随意使用 **kill -9** 而生成，请参阅 *kill*(1) ），则不会将用于共享内存和信号量的系统资源返回给系统。如果必须终止对 **ftio** 的调用，请使用 **kill -15** 。使用 **ipcs** 命令（请参阅 *ipcs*(1) ），可以检查系统当前使用共享内存和信号量的情况。使用 **ipcrm** （请参阅 *ipcrm*(1) ）可以删除已提交的资源。

作者

ftio 由 HP 开发。

另请参阅

cpio(1)、 find(1)、 ipcs(1)、 ipcrm(1)、 kill(1)、 ls(1)、 rmt(1M)、 mknod(2)、 prealloc(2)、 signal(2)、 uname(2)、 acl(5)、 environ(5)、 lang(5)、 regexp(5)、 mt(7)。

名称

ftp - 文件传输程序

概要

ftp [**-g**] [**-i**] [**-n**] [**-c**] [**-v**] [**-p**] [**-P**] [**-l**] [**-B** *size*] [*server-host*]

说明

ftp 是文件传输协议的用户接口。 **ftp** 通过网络连接在本地“客户端”主机与远程“服务器”主机之间复制文件。 **ftp** 在客户端主机上运行。

选项

ftp 命令支持下列选项:

-g 禁用文件名“匹配替换”功能；请参阅下面的 **glob** 命令。缺省情况下，如果未指定该选项，将启用文件名匹配替换功能。

-i 禁用由多文件命令给出的交互式提示；请参阅下面的 **prompt** 命令。缺省情况下，如果未指定该选项，将启用提示功能。

-n 禁用“自动登录”功能；请参阅下面的 **open** 命令。缺省情况下，如果未指定该选项，将启用自动登录功能。

-c 如果设置了该选项，则建立连接时 **ftp** 客户端不会对服务器执行 **SYST** 和 **TYPE** 调用。 **-c** 选项仅在禁用自动登录时才有效。也就是说，该选项在与 **-n** 选项一起被调用时有效。该选项并不禁用 **SYST** 和 **TYPE** 命令，而只是限制了建立连接时对这些命令的调用。

-v 启用详细输出信息；请参阅下面的 **verbose** 命令。如果未指定该选项，则 **ftp** 仅在标准输入与终端相关联时才显示详细输出。

-p 启用操作的被动模式。另请参阅下面的命令部分中的 **passive** 命令。如果未指定该选项，缺省为禁用被动模式。

-P 禁用 Kerberos 身份验证与授权。仅适用于基于 Kerberos V5 的安全环境。指定该选项时，要求提供口令，且该口令会以一种可读的格式跨越网络发送。缺省情况下，如果未指定该选项，也就不需要提供口令，改为进行 Kerveros 身份验证与授权。请参阅 *sis*(5) 。

-l 在 IPv6 环境中启用 LPRT 和 LPSV 命令，以进行数据连接。在此环境中， **ftp** 缺省使用 EPRT 和 EPSV。在 IPv4 环境中，则使用 PORT 和 PASV 命令。

-B 将数据套接字的缓冲区大小设置为采用 1024 字节格式的大小为 *size* 的块。 *size* 的有效值范围为从 1 到 2097151 之间的整数（缺省值为 56）。

注释：较大的缓冲区大小可以提高快速链接（例如 FDDI）上的 **ftp** 性能，但可能会导致慢速链接（例如 X.25）的连接时间非常长。

注释：如果需要将缓冲区大小设置为 1024 字节倍数以外的其他值，请在 *size* 后紧接“B”，中间不留空格。 *size* 值以字节为单位。例如，要将缓冲区大小设置为值“1500”，请使用 **-B**

1500B。

与 **ftp** 通信的服务器主机的名称可在命令行中设置。如果指定了服务器主机， **ftp** 会立即打开一个到服务器主机的连接；请参阅下面的 **open** 命令。否则， **ftp** 等待用户输入命令。

fallback 选项可在 appdefaults 部分中的 **krb5.conf** 文件内进行设置。有关 appdefaults 部分的详细信息，请参阅 *krb5.conf*(4) 联机帮助页。如果 **fallback** 设置为真，而 kerberos 身份验证失败，则 **ftp** 将使用不安全的身份验证模式。

注释：命令行选项会覆盖配置文件选项。

文件传输协议为 *type* 、 *mode* 、 *form* 和 *struct* 指定文件传输参数。 **ftp** 支持的文件传输协议 *types* 有 **ASCII** 、 **binary** 和 **tenex** 。 **ASCII** 为缺省 FTP *type* （应注意，尽管如此，当 **ftp** 在两个类似系统之间建立连接时，它会自动切换为更有效的 **binary** 类型）。 **ftp** 仅支持文件传输参数的缺省值，其中 *mode* 的缺省值为 **stream** 、 *form* 的缺省值为 **non-print** 、 *struct* 的缺省值为 **file** 。

命令

ftp 支持以下命令。带有嵌入式空格的命令参数必须用引号括起（例如 "argument with embedded spaces"）。

![*command* [*args*]]
在本地主机上调用 Shell。 **SHELL** 环境变量指定要调用的 Shell 程序。如果 **SHELL** 未定义， **ftp** 将调用 **/usr/bin/sh** 。如果指定了 *command* ，则 Shell 将执行它，然后返回到 **ftp** 。否则，将调用交互式 Shell。当 Shell 终止时，将返回到 **ftp** 。

$ *macro-name* [*args*]
执行由 **macdef** 命令定义的宏 *macro-name* 。参数将被传递到未经文件名匹配替换的宏。

account [*passwd*]
提供远程系统所需的补充口令，以便在登录成功完成后立即访问资源。如果不包含任何参数，则提示用户在无回显输入模式下输入帐户口令。

append *local-file* [*remote-file*]
将 *local-file* 复制到 *remote-file* 的结尾处。如果未指定 *remote-file* ，则在使用任何 *ntrans* 或 *nmap* 设置对远程文件名进行更改后，在远程文件的命名中将使用本地文件名称。

ascii 将文件传输 *type* 设置为网络 ASCII。这是缺省类型。

bell 每一文件传输完成后，都发出声音提示。

binary 将文件传输 *type* 设置为 **binary** 。

bye 如果打开了一个连接，则关闭与服务器主机的连接并退出。键入文件结束标志 (EOF) 字符同样可以终止并退出会话。

case 在 **mget** 命令执行过程中切换远程计算机文件名的大小写。 **case** 设置为打开（缺省为关闭）时，远程计算机上文件名称中所有的大写字母都切换为小写字母写入本地目录。

cd *remote-directory*
将服务器主机上的工作目录设置为 *remote-directory* 。

cdup 将服务器主机上的工作目录设置为当前远程工作目录的父目录。

chmod *mode file-name*
将远程系统上文件 *file-name* 的权限模式更改为 *mode* 。

close 终止与服务器主机的连接。使用 **close** 命令不会退出 **ftp** 。将清除所有已定义的宏。

cr 在检索 **ascii** 类型文件的过程中切换回车的去除。在 **ascii** 类型文件传输过程中，记录是用回车/换行序列表示的。当 **cr** 设置为打开（缺省值）时，将从该序列中去除回车，以符合 UNIX 的单个换行记录分隔符的规范。非UNIX 远程系统上的记录可以包含单个换行符；在进行 **ascii** 类型传输时，仅当 **cr** 设置为关闭时，才能将记录分隔符与这些换行符区分开来。

delete *remote-file*
删除 *remote-file* 。 *remote-file* 可以是空目录。未进行任何文件名匹配替换。

dir [*remote-directory*] [*local-file*]
将 *remote-directory* 列表写入标准输出，也可以选择写入 *local-file* 。如果 *remote-directory* 和 *local-file* 均未指定，则在标准输出中列出远程工作目录。如果打开了交互式提示，则 **ftp** 会提示用户验证最后一个参数是否确实为 **dir** 输出的目标文件。文件名匹配替换的字符始终被扩展。

disconnect
与 **close** 同义。

longaddr
为 IPv6 环境中的数据连接切换 **LPRT/LPSV** 命令的使用。缺省情况下，在 IPv6 环境中使用 **EPRT/EPSV** 命令。在 IPv4 环境中，则使用 **PORT/PASV** 命令。

form *format*
将文件传输 *form* 设置为 *format* 。支持的唯一格式为 **non-print**

get *remote-file* [*local-file*]
将 *remote-file* 复制到 *local-file* 。如果未指定 *local-file* ，则 **ftp** 将使用指定的 *remote-file* 名称作为 *local-file* 名称，这受当前的 *case* 、 *ntrans* 和 *nmap* 设置所进行的修改的约束。

glob 切换文件名匹配替换。如果启用了文件名匹配替换，则 **ftp** 将扩展文件名和目录名中的 *csh*(1) 元字符。这些字符包括： ***** 、 **?** 、 **[** 、 **]** 、 **~** 、 **{** 和 **}** 。服务器主机扩展远程文件名和目录名。对于 **ls** 和 **dir** 命令，始终扩展文件名匹配替换元字符。如果启用了文件名匹配替换，对于多文件命令 **mdelete** 、 **mdir** 、 **mget** 、 **mls** 和 **mput** ，也会扩展元字符。

hash 对于传输的每 1024 字节，切换 # 字符 (**#**) 的输出。请注意，使用这一功能可能会导致系统性能的降低。

help [*command*]
输出有关名为 *ftp-command* 的 **ftp** 命令的信息消息。如果未指定 *ftp-command* ，则输出所有 **ftp** 命令的列表。

idle [*seconds*]

将远程服务器上的非活动计时器设置为 *seconds* 秒。如果省略 *seconds* ，则 **ftp** 输出当前非活动计时器。

lcd [*local-directory*]

将本地工作目录设置为 *local-directory* 。如果未指定 *local-directory* ，则将本地工作目录设置为用户的本地主目录。

ls [*remote-directory*] [*local-file*]

将 *remote-directory* 列表写入 *local-file* 。该列表列出了服务器选择包含的所有系统相关信息。例如，大多数 UNIX 系统（包括 HP-UX）利用 **ls -l** 命令生成输出（另请参阅 **nlist** ）。如果 *remote-directory* 和 *local-file* 均未指定，则列出远程工作目录。如果启用了文件名匹配替换功能，则将扩展文件名匹配替换的元字符。

macdef *macro-name*

定义一个宏。后续行将存储为宏 *macro-name* ；空输入行将终止宏输入模式。有一个限制，即在所有定义的宏中，宏不得超过 16 个，总字符数不得超过 4096。在执行 **close** 命令之前，宏保留定义状态。宏处理程序将 **$** 和 **** 解释为特殊字符。后跟一个或多个数字的 **$** ，在宏调用命令行上将替换为相应的参数。后跟 *i* 字符的 **$** ，对宏处理程序来说就表示循环执行该宏。第一次循环时， **$***i* 被替换为宏调用命令行中的第一个参数，第二次循环时被替换为第二个参数，依此类推。后跟任意字符的 **** ，将被替换为该字符本身。使用 **** 可防止对 **$** 进行特殊处理。

mdelete [*remote-files*]

删除 *remote-files* 。如果启用了文件名匹配替换功能，则将扩展文件名匹配替换的元字符。

mdir *remote-files local-file*

将 *remote-files* 列表写入 *local-file* 。如果启用了文件名匹配替换功能，则将扩展文件名匹配替换的元字符。如果打开了交互式提示，则 **ftp** 会提示用户验证最后一个参数是否确实为 **mdir** 输出的目标本地文件。

mget *remote-files*

将 *remote-files* 复制到本地系统中。如果启用了文件名匹配替换功能，则将扩展文件名匹配替换的元字符。根据 *case* 、 *ntrans* 和 *nmap* 设置，对得到的本地文件名进行处理。

mkdir *directory-name*

创建远程 *directory-name* 。

mls *remote-files local-file*

将 *remote-files* 的简写列表写入 *local-file* 。如果启用了文件名匹配替换功能，则将扩展文件名匹配替换的元字符。如果 打开了交互式提示，则 **ftp** 会提示用户验证最后一个参数是否确实为 **mls** 输出的目标本地文件。

mode [*mode-name*]

将 FTP 文件传输 *mode* 设置为 *mode-name* 。支持的唯一模式为 **stream** 。

modtime *remote-file*

显示 *remote-file* 的上次修改时间。

mput *local-files*

将 *local-files* 从本地系统复制到远程系统。根据 *ntrans* 和 *nmap* 设置，对与本地文件同名的远程文件进行处理。如果启用了文件名匹配替换功能，则将扩展文件名匹配替换的字符。

newer *file-name*

仅当远程文件的修改时间晚于当前系统上该文件的修改时间，才获取该文件。如果当前系统上不存在该文件，则将远程文件视为 *newer* 。在其他方面，该命令等同于 **get** 命令。

nlist [*remote-directory*] [*local-file*]

将 *remote-directory* 的简写列表写入 *local-file* 。如果未指定 *remote-directory* ，则使用当前工作目录。如果打开了交互式提示，则 **ftp** 会提示用户验证最后一个参数是否确实为 **nlist** 输出的目标本地文件。

nmap [*inpattern outpattern*]

设置或取消文件名映射机制。如果未指定任何参数，则取消文件名映射机制。如果指定了参数，则在执行 **mput** 命令和 **put** 命令且不指定远程目标文件名时，映射远程文件名。如果指定了参数，则在执行 **mget** 命令和 **get** 命令且不指定本地目标文件名时，映射本地文件名。当连接到使用不同的文件命名约定或惯例的非UNIX 远程计算机时，该命令十分有用。映射所遵循的模式由 *inpattern* 和 *outpattern* 进行设置。 *inpattern* 是一种供传入文件名（可能已根据 **ntrans** 和 **case** 设置进行了处理）使用的模板。变量的模板化是通过在 *inpattern* 中包含序列 **$1** 、 **$2**、 ... 、 **$9** 而实现的。使用 **** 可以防止对 **$** 字符进行该特殊处理。所有其他字符将按字面意思进行处理，并用于确定 **nmap** *inpattern* 变量值。例如，假定 *inpattern* 为 **$1.$2** ，远程文件名为 **mydata.data** ，则 **$1** 的值为 **mydata** ， **$2** 的值为 **data** 。 *outpattern* 确定所得到的映射文件名。序列 **$1** 、 **$2**、 ... 、 **$9** 将被替换为从 *inpattern* 模板得到的任意值。序列 **$0** 将被替换为原始文件名。此外，如果 *seq1* 不是空字符串，则序列 **[***seq1***,***seq2***]** 将被替换为 *seq1* ；否则，将被替换为 *seq2* 。例如，命令 **nmap $1.$2.$3 [$1,$2].[$2,file]** 可为输入文件名 **myfile.data** 和 **myfile.data.old** 生成输出文件名 **myfile.data** ，为输入文件名 **myfile** 生成输出文件名 **myfile.file** ，为输入文件名 **.myfile** 生成输出文件名 **myfile.myfile** 。 *outpattern* 中可以包含空格，例如： **nmap $1 | sed "s/ *$//" > $1**. 使用 **** 字符可防止对 **$** 、 **[** 、 **]** 和 **,** 进行特殊处理。

ntrans [*inchars* [*outchars*]]

设置或取消文件名字符转换机制。如果未指定任何参数，则取消文件名字符转换机制。如果指定了参数，则在执行 **mput** 命令和 **put** 命令且不指定远程目标文件名时，转换远程文件名中的字符。如果指定了参数，则在执行 **mget** 命令和 **get** 命令且不指定本地目标文件名时，转换本地文件名中的字符。当连接到使用不同的文件命名约定或惯例的非UNIX 远程计算机时，该命令十分有用。文件名中与 *inchars* 中的某个字符匹配的字符，将被替换为 *outchars* 中的相应字符。如果字符在 *inchars* 中的位置长于 *outchars* 的长度，则从文件名中删除该字符。

open *server-host* [*port-number*]

使用 *port-number* （如果已指定）建立与 *server-host* 的连接。如果启用了 *auto-login* ，则 **ftp** 会尝试登录到服务器主机。

passive 切换传输的被动模式。缺省情况下，将禁用传输的被动模式。该命令使服务器可以指定 FTP 传输所使用的数据端口。

prompt
切换交互式提示。缺省情况下，在执行多文件命令过程中 **ftp** 会提示用户为每个输出文件给出是或否的响应。如果禁用了交互式提示， **ftp** 将对所有指定文件执行命令。

proxy *ftp-command*
在次控制连接上执行 **ftp** 命令。该命令允许同时连接到两个远程 FTP 服务器，以便在这两个服务器之间传输文件。第一个 **proxy** 命令应该是 **open** 命令，用于建立次控制连接。输入 **proxy ?** 命令，可查看能在该次连接上执行的其他 FTP 命令。下列命令以 **proxy** 开头时，其行为与原来完全不同： **open** 在自动登录进程中不定义新宏； **close** 不清除现有的宏定义； **get** 和 **mget** 将文件从主控制连接上的主机传输到次控制连接上的主机； **put** 、 **mput** 和 **append** 将文件从次控制连接上的主机传输到主控制连接上的主机。第三方文件的传输取决于次控制连接上的服务器是否支持 FTP 协议的 **PASV** 命令。

put *local-file* [*remote-file*]
将 *local-file* 复制到 *remote-file* 。如果未指定 *remote-file* ，则 **ftp** 会将 *local-file* 名称（已根据 *ntrans* 或 *nmap* 设置进行处理）指定给 *remote-file* 名称。

pwd 将远程工作目录的名称写入 *stdout* 。

quit 与 **bye** 同义。

quote *arguments*
将 *arguments* 逐字地发送到服务器主机。请参阅 *ftpd*(1M) 。

recv *remote-file* [*local-file*]
与 **get** 同义。

reget *remote-file* [*local-file*]
reget 与 **get** 命令类似，不同之处在于，如果 *local-file* 存在且比 *remote-file* 小，则假定 *local-file* 是 *remote-file* 的不完全传输副本，并从显而易见的失败点处继续传输。在通过经常断线的网络传输大文件时，该命令非常有用。

rhelp [*command-name*]
请求服务器主机的帮助。如果指定了 *command-name* ，则将其提供给服务器。请参阅 *ftpd*(1M) 。

rstatus [*file-name*]
如果不带任何参数，则显示远程计算机的状态。如果指定了 *file-name* ，则显示远程计算机上 *file-name* 的状态。

rename *remote-from remote-to*
将 *remote-from* （它可以是文件或目录）重命名为 *remote-to* 。

reset 清除响应队列。该命令可使命令/响应序列与远程 FTP 服务器重新同步。在远程服务器违背 FTP 协议后，必须进行重新同步操作。

restart *marker*

在指定的 *marker* 处重新启动紧跟其后的 **get** 或 **put** 命令。在 UNIX 系统中，标记通常是文件中的字节偏移量。

rmdir *remote-directory*

删除 *remote-directory* 。 *remote-directory* 必须是空目录。

runique

切换本地系统上使用唯一文件名的文件存储。如果某个现有文件的文件名与 **get** 或 **mget** 命令要获得的目标本地文件名相同，则在该名称后追加 **.1** 。如果得到的名称与另一个现有文件匹配，则在原始名称后追加 **.2** 。如果此进程一直持续到 **.99** ，则输出错误消息，且不进行传输。 **ftp** 将报告唯一的文件名。请注意， **runique** 并不影响通过 Shell 命令生成的本地文件（请参阅下文）。缺省值为 **off** 。

send *local-file* [*remote-file*]

与 **put** 同义。

sendport

切换 **PORT** 命令的使用。缺省情况下，当为每次数据传输建立连接时， **ftp** 都会尝试使用 **PORT** 命令。如果 **PORT** 命令失败， **ftp** 将使用缺省的数据端口。如果禁用 **PORT** 命令，在每次进行数据传输时， **ftp** 不会尝试使用 **PORT** 命令。某些 FTP 实现忽略 **PORT** 命令，但错误地指明已接受这些命令，在这种情况下该命令非常有用。请参阅 *ftpd*(1M) 。关闭 **sendport** 可能会导致命令执行的延迟。

site *arguments*

将 *arguments* 作为 **SITE** 命令逐字地发送到服务器主机。请参阅 *ftpd*(1M) 。

size *remote-file*

显示 *remote-file* 的大小。

status 显示 **ftp** 的当前状态。

struct [*struct-name*]

将 FTP 文件传输 *struct* 设置为 *struct-name* 。支持的唯一 *struct* 为 **file** 。

sunique

切换远程计算机上使用唯一文件名的文件存储。远程服务器将报告唯一的名称。缺省情况下， **sunique** 为 **off** 。

system 显示远程计算机上所运行的操作系统的类型。

tenex 将 FTP 文件传输 *type* 设置为 **tenex** 。

type [*type-name*]

将 FTP 文件传输 *type* 设置为 *type-name* 。如果未指定 *type-name* ，则将当前 *type* 写入 *stdout* 。当前支持的 *types* 包括 **Ascii** 、 **binary** 和 **tenex** 。

umask [*newmask*]

将远程服务器上的缺省 umask 设置为 *newmask* 。如果省略了 *newmask* ，则输出当前 umask。

user *user-name* [*password*] [*account*]

登录到当前连接上的服务器主机，该服务器主机必须已打开。用户本地主目录中的 **.netrc** 文件可以提供 *user-name* 、 *password* 和可选的 *account* ；请参阅 *netrc*(4) 。否则， **ftp** 将提示用户输入该信息。 HP-UX FTP 服务器不需要 *account* 。出于安全原因， **ftp** 始终需要口令。它不会登录到没有口令的远程帐户。

在基于 Kerberos V5 的安全环境中， **ftp** 不需要口令，而是改为执行 Kerveros 身份验证与授权，如 *sis*(5) 中所述。在所有其他环境中，如果用户提供了正确的口令，则认为该用户已通过身份验证；如果该用户在远程系统中有相应的 *account* ，则认为该用户已被授权。

verbose

切换详细输出。如果启用了详细输出，则 **ftp** 将显示服务器主机的响应，当文件传输完成时，它将报告有关传输效率的统计信息。

? [*command*]

与 **help** 命令同义。对于指定的 *command* ，输出 **help** 信息。

中止文件传输

要中止文件传输，请使用终端中断键（通常为 **Ctrl-C** ）。文件传输的发送过程会立即停止。通过先向远程服务器发送 FTP 协议的 **ABOR** 命令，然后忽略所有进一步接收的数据， **ftp** 停止文件传输的传入（接收）过程。这个过程的完成速度取决于远程服务器对 **ABOR** 处理的支持情况。如果远程服务器不支持 **ABOR** 命令，则一直到远程服务器发送完所请求的文件之后，才会显示提示符 **ftp>** 。

当 **ftp** 等待远程服务器的响应时，终端中断键序列会被忽略。在这种模式下，如果出现如上所述的 **ABOR** 处理，或远程服务器的异常行为（包括违反 FTP 协议），则会导致较长的延迟。如果延迟是由远程服务器的异常行为造成的，则必须手动终止本地的 **ftp** 程序。

文件命名约定

指定为 **ftp** 命令参数的文件，将根据以下规则进行处理。

- 如果指定了文件名 **-** ，则 **ftp** 使用标准输入（用于读取）或标准输出（用于写入）。
- 如果文件名的第一个字母为 **|** ，则 **ftp** 将参数的剩余部分解释为 Shell 命令。 **ftp** 使用 **popen()** （请参阅 *popen*(3S) ）及提供的参数派生 Shell，然后从标准输出（标准输入）进行读取（写入）。如果 Shell 命令中包含空格，则必须用引号将参数括起，例如：

  "| ls -lt".

  下面是该机制的几个有用示例：

  ls . "| more".

  上述命令将逐页列出当前目录中的文件。

  put "| tail -20 loc_file" rem_file.

该命令会将本地文件“loc_file”中的最后 20 行作为“rem_file”复制到远程系统中。

- 另外，如果启用了文件名匹配替换功能， **ftp** 会根据 C Shell 使用的规则扩展本地文件名（请参阅 *csh*(1) ）；请参阅下面的 **glob** 命令。如果 **ftp** 命令需要一个本地文件（例如 **put** ），则仅使用文件名匹配替换操作所生成的第一个文件名。
- 对于 **mget** 命令和 **get** 命令而言，如果未指定本地文件名，则将远程文件名用作本地文件名，可通过 **case** 、**ntrans** 或 **nmap** 设置对该文件名进行更改。如果 **runique** 处于打开状态，则所得到的文件名随后可能会被更改。
- 对于 **mput** 命令和 **put** 命令而言，如果未指定远程文件名，则将本地文件名用作远程文件名，可通过 **ntrans** 或 **nmap** 设置对该文件名进行更改。如果 **sunique** 处于打开状态，则所得到的文件名随后可能会被远程服务器更改。

警告

许多命令的正确执行取决于远程服务器的行为适当与否。

诊断信息

Error! could not retrieve authentication type.

Please notify sys admin.

ftp 使用两种身份验证机制。一种身份验证机制是以 Kerberos 为基础的，而另外一种则不是。身份验证机制的类型是通过由 **inetsvcs_sec** 更新的系统文件获得的（请参阅 *inetsvcs_sec*(1M) ）。如果系统文件不包含已知的身份验证类型，则会显示上述错误。

作者

ftp 由加州大学伯克利分校开发。

另请参阅

csh(1)、 rcp(1)、 ftpd(1M)、 inetsvcs_sec(1M)、 ftpusers(4)、 hosts(4)、 krb5.conf(4)、 netrc(4)、 sis(5)。

名称

ftpcount - 显示每个级别的当前用户数

概要

/usr/bin/ftpcount [**-V**]

说明

ftpcount 命令显示在 ftpaccess 文件中定义的每个级别的当前用户数（以及限制）。如果 **ftpaccess** 文件不存在，**ftpcount** 命令将不显示任何内容。然而，如果 **ftpaccess** 文件存在并且占用零个字节， **ftpcount** 将显示一条错误消息：

ftpcount:no service classes defined, no usage count kept.

-V 选项会使程序显示版权和版本信息，然后终止。

退出状态

ftpcount 返回：

0 成功

1 失败

作者

ftpcount 由密苏里州华盛顿大学圣路易斯分校开发。

另请参阅

ftpwho(1)、 ftpaccess(4)。

名称

ftprestart - 删除 ftpshut 创建的关闭消息文件

概要

/usr/bin/ftprestart [**-V**]

说明

ftprestart 命令会从真实、匿名和虚拟用户帐户中删除所有关闭消息文件。该消息文件由 **ftpshut** 实用程序创建，该实用程序位于 **/etc/ftpd/ftpaccess** 文件中的 **shutdown** 指令指定的路径中（有关详细信息，请参阅 *ftpshut*(1)）。该命令总是在执行 **ftpshut** 命令后使用。

Note: 对于 *guest* 用户帐户，必须手动删除这些消息文件。执行 **ftprestart** 命令不会完成删除操作。 **-V** 选项使程序显示版权和版本信息，然后终止。

返回值

ftprestart 将其退出状态设置为:

0 成功

1 失败

举例

输出示例

ftprestart: /servers/some.domain/ftp/etc/shutmsg removed.

ftprestart: /servers/other.domain/ftp/etc/shutmsg removed.

ftprestart: /etc/shutmsg removed.

作者

ftprestart 由密苏里州华盛顿大学圣路易斯分校开发。

另请参阅

ftpshut(1)、 ftpaccess(4)。

名称

ftpshut - 创建关闭消息文件，以便在指定时间关闭 FTP 服务器

概要

/usr/bin/ftpshut [**-V**] [**-l** *min*] [**-d** *min*] *time* [*warning-message*]

说明

ftpshut 命令提供一个自动关闭程序，在关闭 FTP 服务器时，超级用户可以用它来通知 FTP 用户。该命令将创建一个关闭消息文件，创建文件所在的文件路径是使用实际的匿名帐户和虚拟用户帐户中 **/etc/ftpd/ftpaccess** 文件中的 **shutdown** 指令指定的。对于来宾帐户，系统管理员必须将实际用户帐户中创建的消息文件手动复制到来宾帐户。服务器将定期检查该文件，以查看是否将要关闭服务器。

选项

-d *min* 设置断开偏移量。所有当前的 FTP 连接都将在关闭前断开 *min* 分钟。 *min* 的缺省值为 5 分钟。用户可以重置该值。

-l *min* 设置拒绝偏移量。在关闭之前禁用新 FTP 访问 *min* 分钟。 *min* 的缺省值为 10 分钟。用户可以重置该值。

-V 显示版权和版本信息，然后终止。

操作数

time 将关闭 FTP 服务器的时间。如果将 **time** 设置为词 **now** ，则会立即关闭。还可以将 *time* 设置为将来的时间。可以使用以下两种格式中的任何一种来指定将来时间： **+***number* 或 *HHMM* 。第一种格式在 *number* 分钟后关闭 FTP 服务器。第二种格式将在一天指定的时间关闭 FTP 服务器，使用 24-小时时钟格式。

warning-message

服务器在关闭时将在其客户端闪烁显示的消息。用户可以使用自己选择的消息，或者使用可用的“宏”或“Magic Cookie”。服务器将使用指定的文本字符串来代替宏。*warning-message* 将格式化为 75 个字符长，包括任何扩展的宏（“Magic Cookie”）的长度。缺省的警告消息为“**System shutdown at** *%s*”。可使用以下 Magic Cookie :

%s 系统将要关闭的时间

%r 将拒绝新连接的时间

%d 将断开当前连接的时间

%C 当前工作目录

%E 维护人员的电子邮件地址（如 **ftpaccess** 中所定义）

%L 本地主机名

%M 该类中最多允许的用户数

%N 该类中当前用户数

%R 远程主机名

%T 本地时间（形式：Thu Nov 15 17:12:42 1990）

%U 登录时指定的用户名

返回值

ftpshut 将其退出值设置为：

0 成功。

1 失败。

-1 将错误的参数传递到 **ftpshut** 。

警告

如果您为 **ftpshut** 使用绝对时间，则您只能在现在和 23:59 之间关闭服务器。

作者

ftpshut 由密苏里州华盛顿大学圣路易斯分校开发。

另请参阅

ftprestart(1)、 ftpaccess(4)。

名称

ftpwho - 显示每个 ftp 用户的当前进程信息。

概要

/usr/bin/ftpwho [**-V**]

说明

ftpwho 命令显示登录到 ftp 服务器的每个用户的当前进程信息。如果 **ftpaccess** 文件不存在，该命令将不显示任何信息。然而，如果 ftpaccess 文件存在且占用零个字节，则该命令将显示一条错误消息：

ftpwho: no service classes defined, no usage count kept.

-V 选项可使程序显示版权和版本信息，然后终止。

退出状态

ftpwho 返回：

0 成功

1 失败

作者

ftpwho 由密苏里州华盛顿大学圣路易斯分校开发。

另请参阅

ftpcount(1)、ftpaccess(4)。

名称

gencat - 生成格式化的消息清单文件

概要

gencat [**-l**] *catfile msgfile* ...

说明

通过消息清单，程序可以按照本地的惯例和语言处理输入并生成输出。有关详细信息，请参阅《Native Language Support Users Guide》。

gencat 命令将每个消息源 *msgfile* 合并到一个格式化的消息清单 *catfile* 中，它可以通过 **catgets()** 进行访问（请参阅 *catgets*(3C)）。如果不存在 *catfile*，则将其创建。如果存在 *catfile*，其消息将包括在新的 *catfile* 中。如果集合号与消息号发生冲突，*file* 中的新消息文本将替换 *catfile* 中的旧消息文本。*msgfile* 由下述的消息、指令和注释行组成（均不带前导空格或制表符）。除非另行注明，字段将由一个或多个空格或制表符进行分隔。

如果 **-** 指定为清单文件，则将使用标准输出。

如果为消息文件的实例指定 **-**，则使用标准输入。

$set *s* [*comment*]
: **$set** 指令指定下一个 **$set** 或文件结束标志之前的消息的集合 *s*。集合号 *s* 是无符号的整数，范围从 1 到 **NL_SETMAX**。集合号后的任何字符串均作为注释进行处理。如果未指定 **$set** 指令，则消息将放入缺省集合 **NL_SETD**。

  在 *msgfile* 中，集合号必须按升序排列，但不必是连续的。

$delset *s* [*comment*]
: **$delset** 指令在现有消息清单中删除由集合号 *s* 标识的消息集合。集合号后的任何字符串均作为注释进行处理。

m message_text
: 消息行指定消息号 *m* 和关联的消息文本。消息号 *s* 是无符号的整数，范围从 1 到 **NL_MSGMAX**。*message_text* 是 C 字符串，包括空格、制表符和\（反斜杠）转义符，但在缺省情况下不带有引号（请参阅下文的 **$quote** 指令）。消息号 *m* 通过单个空格或制表符与 *message_text* 分隔。*message_text* 以分隔符后的第一个字符开头，在换行符处结尾。额外的空格或制表符（包括任何尾随的空格或制表符）将被视作 *message_text* 的一部分。

  消息行的 *message_text* 以最近的 **$set** 指令所指定的消息号 *m* 和集合号 *s* 存储在 *catfile* 中。

  消息号必须在集合中按升序排列，但不必是连续的。

  请注意，空格或制表符分隔符可区分空消息的插入和消息的删除。如果消息行包含编号和分隔符，但不包含文本，消息号和关联的空消息字符串将存储在 *catfile* 中。如果消息行包含编号，但不包含分隔符和文本，消息号和关联的空消息字符串将从 *catfile* 中删除。

-l
: 如果指定了 **-l** 选项，*message_text* 的长度不得大于 **MAX_BUFLEN** – 1 字节。如果未指定 **-l** 选项，*message_text* 的长度不得大于 **NL_TEXTMAX** 字节。有

关这些例程所强加的消息长度限制的信息，请参阅 *catgets*(3C) 。

$quote [*q comment*] **$quote** 指令指定用于给 *message_text* 添加引号的引号符 *q* ，并使得前导和尾随的空格在消息行中可见。指定引号符 *q* 后的任何字符串均作为注释进行处理。缺省情况下或者在未提供引号符 *q* 时，将不识别 *message_text* 的引用。

$ *comment* 空格或制表符后的 **$** 作为注释进行处理，并且可以出现在文件中的任意位置。由零个或更多个空格或制表符组成的行作为注释行进行处理。

NL_TEXTMAX 、 **NL_SETMAX** 和 **NL_MSGMAX** 在 **<limits.h>** 中定义。 **NL_SETD** 在 **<nl_types.h>** 中定义。 **MAX_BUFLEN** 在 **<msgcat.h>** 中定义。

外部语言环境影响

环境变量

LANG 为没有设置或设置为空的国际化变量提供了缺省值。如果 **LANG** 未设置或设置为空，则会使用缺省值“C”（请参阅 *lang*(5) ）。如果任一国际化变量中包含无效设置，则 **gencat** 的行为类似于所有国际化变量都设为“C”。请参阅 *environ*(5) 。

LC_ALL 如果设为非空字符串值，则会覆盖所有其他国际化变量的值。

LC_CTYPE 确定作为单字节和（或）多字节字符的文本的解释、作为可打印字符的字符分类、以及在正则表达式中按字符类表达式匹配的字符。

LC_MESSAGES 确定语言环境，该语言环境影将用于改变写入到标准错误的诊断消息以及写入到标准输出的信息消息的格式和内容。

NLSPATH 用于确定消息清单的位置，以便处理 **LC_MESSAGES** 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

警告

HP MPE 和 RTE 操作系统目前不支持 **$quote** 指令。

作者

gencat 由 HP 和 X/Open Company, Ltd. 联合开发。

另请参阅

dumpmsg(1)、 findmsg(1)、 insertmsg(1)、 catgets(3C)、catopen(3C)。

《Native Language Support Users Guide》

符合的标准

gencat： XPG2、XPG3

名称

genxlt - 生成 iconv 转换表

概要

genxlt [**-f** *output_filename*] [*input_filename*]

说明

genxlt 生成 iconv 表的已编译、不可读的二进制版本，该二进制版本适用于 *iconv*(1) 和 *iconv*(3C) 。如果未指定 *input_filename* 或 *output_filename* ，则使用标准输入和/或标准输出。

由于 **genxlt** 的输出为二进制的不可读文件，因此，如果未使用 **-f** 选项，则可以使用重定向符号 **>** 将标准输出重定向到一个文件中。

选项

genxlt 可识别下列选项：

-f *output_filename*
: 如果未选定此选项，则将数据发送到标准输出，在标准输出中可将数据重定向到一个文件。

genxlt 创建指定格式的表， *iconv*(3C) 的缺省转换例行程序可以对该表进行解释。输入文件有两列，提供两个代码集之间的文件代码映射。条目以十六进制表示。

必须将输入文件格式化为两列十六进制数字。将第一列中的字符转换为第二列中的字符。将第一列中前面带有 **#** 的行被忽略，作为所有行中的注释，但下列关键字的情况除外。 *#Galley:* 和 *#What:*

除了定义文件代码映射的数据以外，还可以在特定转换中定义预定义 (*Galley*) 字符（请参阅 *iconv*(3C) ）。通过将 **#Galley: 0xnnnn** 行添加到输入文件的开头，可实现此操作。 **nnnn** 为多字节字符（请参阅举例）。也可以使用条目 **#What:** <*any_string*> ，在输入文件中定义 *What* 字符串（请参阅 *what*(1) ）。此字符串可以包含如版本号、转换类型等信息，这些信息不能以任何方式用于转换。请注意，如果定义了 *What* 字符串，则它应出现在 *Galley* 定义之前。

外部语言环境影响

环境变量

LANG 为没有设置或设置为 null（空）的国际化变量提供了缺省值。如果 **LANG** 未设置或设置为 null（空），则会使用缺省值“C”（请参阅 *lang*(5) ）。如果任一国际化变量中包含无效值，则 **genxlt** 就会认为所有国际化变量都设置为“C”（请参阅 *environ*(5) ）。

如果将 **LC_ALL** 设置为非空字符串值，则它将覆盖所有其他国际化变量的值。

LC_MESSAGES 确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息以及写入到标准输出的信息消息的格式和内容。

NLSPATH 确定消息目录的位置，以便处理 **LC_MESSAGES** 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

退出值包括:

0 成功完成。

>0 发生了错误。

举例

此示例编译 *iconv_input* ，并将二进制输出结果置于 **/usr/lib/nls/iconv/tables/roma8=iso81** 中。下列 iconv 语句使用 *roma8=iso81* 表将 *data_file* 从 **roman8** 代码集转换为 **iso8859-1** 代码集。

```
% genxlt iconv_input > /usr/lib/nls/iconv/tables/roma8=iso81
% iconv -f roma8 -t iso81 data_file
```

这是输入文件的示例:

```
#What: CodesetA to CodesetB: version 1.0
#Galley:   0xffff
# the conversion data is as follows:
0x01    0x01
0x02    0x42
...
0xff87    0x4567
...
etc.
```

警告

由于 **genxlt** 将覆盖现有表，因此在使用 **genxlt** 之前将现有表保存到另一个文件是非常明智的。

对输入文件中的错误数据提供警告。

必须具有超级用户权限，才能将文件安装在 **/usr/lib/nls/iconv/tables** 中。

文件

/usr/lib/nls/iconv/tables 所有表都必须安装在该目录下。

另请参阅

dmpxlt(1)、 iconv(1)、 iconv(3C)。

符合的标准

genxlt：XPG4 表

名称

get - 获取 SCCS 文件的版本

概要

get [**-r** *SID*] [**-c** *cutoff*] [**-e**] [**-b**] [**-i** *list*] [**-x** *list*] [**-k**] [**-l**[**p**]] [**-p**] [**-s**] [**-m**] [**-n**] [**-g**] [**-t**] [**-w** *string*] [**-a** *seq-number*] *file* ...

说明

get 命令按照其选项参数（以 **-** 开头）指定的规范从每个命名的 SCCS 文件中生成 ASCII 文本文件。参数可以按任意顺序指定，但所有选项参数必须应用于所有命名的 SCCS 文件。如果命名了目录， **get** 将如同目录中的每个文件均指定为命名文件一样进行操作，例外的是将以静默方式忽略非 SCCS 文件（路径名的最后一部分不以 **s.** 开头）和不可读的文件。如果给定了 **-** 的文件名，则将读取标准输入，并且假定标准输入的每一行是要处理的 SCCS 文件的名称。同样，非 SCCS 文件和不可读的文件将以静默方式忽略。

生成的文本通常会写入称作 *g-file* 的文件中，其名称派生自 SCCS 文件名（只需删除 **s.** 前缀即可；请参阅下文介绍的“文件”部分）。

选项

下面的选项参数说明基于仅处理一个 SCCS 文件的情况。当处理多个 SCCS 文件时，所有选项参数的作用都将独立地应用于各命名文件。

-r*SID* 要检索的 SCCS 文件的版本（delta 版）的 *S*CCS *ID*entification 字符串 (SID)。表 1 显示在最有用的情况下作为指定的 SID 的功能而所检索的 SCCS 文件的版本（如果还使用了 **-e** 选项，则包括将最终由 **delta** 创建的版本的 SID）（请参阅 *delta*(1)）。

-c*cutoff* *cutoff* 日期和时间，其格式如下：

YY[*MM*[*DD*[*HH*[*MM*[*SS*]]]]]

对在指定 *cutoff* 日期和时间后创建的 SCCS 文件的更改（delta 版）将不包括在生成的 ASCII 文本文件中。从日期和时间中省略的单位缺省为它们的最大可能值；也就是说 **-c7502** 等效于 **-c750228235959** 。任意数量的非数值字符可以分隔 *cutoff* 日期和时间的各个 2 位数部分。该功能允许指定以下格式的 *cutoff* 日期： **-c77/2/2 9:22:25** 。请注意，这意味着可以将 **%E%** 和 **%U%** 标识关键字（请参阅下文）用于命令中的嵌套 **get**s：

~!get "-c%E% %U%" s.file

-e 指示 **get** 的用途是通过随后使用 **delta** 对 SCCS 文件进行编辑或更改（delta 版）。 **get** 中用于 SCCS 文件特定版本 (SID) 的 **-e** 选项防止其他 **get** 命令对相同的 SID 进行编辑，直至执行 **delta** 或者在 SCCS 文件中设置 **j** （连接编辑）标志（请参阅 *admin*(1)）。始终允许将 **get -e** 同时用于不同的 SID。但是，请注意，只允许一个用户指定并发的 **get -e** （请参阅 *admin*(1)）。

如果在编辑过程中意外地损坏了带 **-e** 选项的 **get** 生成的 *g-file* ，则可以通过以 **-k** 选项代替 **-e** 选项重新执行 **get** 命令来重新生成该文件。

通过存储在 SCCS 文件中的上限、下限和授权用户列表指定的 SCCS 文件保护（请参阅 *admin*(1) ）将在使用 **-e** 选项时强制执行。

-b 与 **-e** 选项一起使用来指示新的 delta 版应该具有新分支中的 SID ，如表 1 所示。如果文件中没有 **b** 标志（请参阅 *admin*(1) ）或者检索的 *delta* 更改量不是叶 *delta* 更改量，则将忽略该选项（叶 *delta* 更改量是在 SCCS 文件树上没有后继对象的更改量）。

注意：分支 *delta* 更改量始终可以从非叶 *delta* 更改量中创建。

-i*list* 将在所生成文件的创建过程中包括的 delta 版的 *list* （强制应用）。该 *list* 使用的语法如下：

list ::= *range* | *list*, *range*
range ::= *SID* | *SID* - *SID*

delta 版的 SID （SCCS 标识符）可以采用表 1 的“指定的 SID”列中的任意形式。部分 SID 将按照表 1 的“检索的 SID”列所示进行解释。请参阅“警告”。

-x*list* 将在所生成文件的创建过程中排除的 delta 版的 *list* （强制不应用）。有关 *list* 格式的信息，请参阅 **-i** 选项。

-k 禁止将检索到的文本中的标识关键字（请参阅下文）替换为它们的值。 **-e** 选项暗含 **-k** 选项的功能。

-l[**p**] 致使 delta 版摘要写入 *l-file* 。如果使用了 **-lp** ，则不创建 *l-file* ； delta 版摘要将在标准输出上写入。有关 *l-file* 的格式，请参阅“文件”。用户必须具有 *s-file* 读取权限才能使用 **-l** 选项。

-p 致使从 SCCS 文件中检索的文本在标准输出上写入。不创建 *g-file* 。通常转到标准输出的所有输出将转到文件描述符 2（标准错误），除非使用 **-s** 选项，此时输出将消失。

-s 禁止通常在标准输出上写入的所有输出。但是，致命错误消息（始终转到文件描述符 2）仍不受影响。

-m 致使从 SCCS 文件中检索的每个文本行在前面添加在 SCCS 文件中文本行前插入的 delta 版的 SID 。格式如下： SID，后接横向制表符，后接文本行。

-n 致使每个生成的文本行前面添加 **%M%** 标识关键字值（请参阅下文）。格式如下：**%M%** 值，后接横向制表符，后接文本行。同时使用 **-m** 和 **-n** 选项时，其格式为：**%M%** 值，后接横向制表符，后接 **-m** 选项生成的格式。

-g 禁止从 SCCS 文件中实际检索文本。它主要用于生成 *l-file* 或验证特定 SID 是否存在。

-t 用于在给定发行版（如 **-r1** ）或发行版和级别（如 **-r1.2** ）中访问最近创建的（“顶级”）delta 版。

-w *string* 在通过 **get** 获取文件时将 *string* 替换为 **@%M%** 的所有实例。

-a*seq-number* 要检索的 SCCS 文件 delta 版（版本）的 delta 版序列号（请参阅 *sccsfile*(4)）。该选项由 **comb** 命令使用（请参阅 *comb*(1)）；它不是通用的选项，因此应该避免使用。如果同时指定 **-r** 和 **-a** 选项，则应使用 **-a** 选项。由于要创建的 delta 版的 SID 可能不同于预期的 SID，因此在将 **-a** 选项与 **-e** 选项一起使用时应小心。 **-r** 选项可以与 **-a** 和 **-e** 选项一起使用来控制要创建的 delta 版的 SID 命名。

对于所处理的每个文件， **get** 将以所访问的 SID 以及从 SCCS 文件中检索到的行数进行响应（在标准输出上）。

如果使用了 **-e** 选项，则要创建的 delta 版的 SID 将出现在所访问的 SID 之后，所生成的行数之前。如果存在多个命名文件，或者命名了目录或标准输入，则将在处理之前输出每个文件名（前面带有换行符）。如果使用了 **-i** 选项，则将在标志“已包含”之后列出所包含的 delta 版。如果使用了 **-x** 选项，则将在标志“已排除”之后列出已排除的 delta 版。

表 **1. SCCS** 标识字符串的确定

指定的 SID*	**–b** 选项已用 %	其他条件	已检索的 SID	要创建的 delta 更改量的 SID
无 %%	否	R 缺省为 mR	mR.mL	mR.(mL+1)
无 %%	是	R 缺省为 mR	mR.mL	mR.mL.(mB+1).1
R	否	R > mR	mR.mL	R.1***
R	否	R = mR	mR.mL	mR.(mL+1)
R	是	R > mR	mR.mL	mR.mL.(mB+1).1
R	是	R = mR	mR.mL	mR.mL.(mB+1).1
R	-	R < mR 且 R 不存在	hR.mL**	hR.mL.(mB+1).1
R	-	发行版中的主干后继对象号 > R 且 R 存在	R.mL	R.mL.(mB+1).1
R.L	否	无主干后继对象	R.L	R.(L+1)
R.L	是	无主干后继对象	R.L	R.L.(mB+1).1
R.L	-	发行版中的主干后继对象 ≥ R	R.L	R.L.(mB+1).1
R.L.B	否	无分支后继对象	R.L.B.mS	R.L.B.(mS+1)
R.L.B	是	无分支后继对象	R.L.B.mS	R.L.(mB+1).1
R.L.B.S	否	无分支后继对象	R.L.B.S	R.L.B.(S+1)
R.L.B.S	是	无分支后继对象	R.L.B.S	R.L.(mB+1).1
R.L.B.S	-	分支后继对象	R.L.B.S	R.L.(mB+1).1

表 **1** 注释

* “R”、“L”、“B”和“S”分别表示 SID 的“发行版”、“级别”、“分支”和“序列”部分；“m”表示“maximum”（最大值）。例如，“R.mL”表示“发行版 R 内的最大级别号”；“R.L.(mB+1).1”表示“发行版 R 内级别 L 的 新分支上的第一个序列号（即最大分支号加一）”。请注意，如果指定 SID 的形式为“R.L”、“R.L.B”或“R.L.B.S”，则每个指定的部分均 必须存在。

** “hR”是低于指定的且 不存在的发行版 R 的最高 现有发行版。

*** 它用于在 新发行版中强制创建 第一个 delta 版。

后继对象。

% 只有当文件中存在 **b** 标志时， **-b** 选项才有效（请参阅 *admin*(1)）。 **-** 条目表示“无关”。

%% 该情况在文件中 不存在 **d** （缺省 SID）标志时适用。如果文件中 存在 **d** 标志，则从 **d** 标志中获取的 SID 将按照它在命令上指定的形式进行解释。因此，表中的其他情况之一将适用。

标识关键字

标识信息插入从 SCCS 文件中检索的文本，其方法是将任意位置出现的 标识关键字替换为它们的值。以下关键字可以在存储 SCCS 文件中存储的文本内使用：

关键字	值
%M%	模块名称：文件中 **m** 标志的值（请参阅 *admin*(1)），如果不存在，则是删除前导 **s.** 的 SCCS 文件名。
%I%	已检索的文本的 SCCS 标识 (SID) (**%R%.%L%.%B%.%S%**)。
%R%	发行版。
%L%	级别。
%B%	分支。
%S%	序列。
%D%	当前日期 (YY/MM/DD)。
%H%	当前日期 (MM/DD/YY)。
%T%	当前日期 (HH:MM:SS)。
%E%	创建最新的已应用 delta 版的日期 (YY/MM/DD)。
%G%	创建最新的已应用 delta 版的日期 (MM/DD/YY)。
%U%	创建最新的已应用 delta 版的时间 (HH:MM:SS)。
%Y%	模块类型： SCCS 文件中 **t** 标志的值（请参阅 *admin*(1)）。
%F%	SCCS 文件名。
%P%	完全限定的 SCCS 文件名。
%Q%	文件中 **q** 标志的值（请参阅 *admin*(1)）。
%C%	当前行号。该关键字用于程序输出的消息，如“这不应该发生”之类的错误。它 不用于在每一行上提供序列号。
%Z%	**what** 可识别的 4 字符字符串 **@(#)** （请参阅 *what*(1)）。
%W%	用于构建 HP-UX 系统程序文件的 *what*(1) 字符串的简写形式。

%W%=%Z%%M%*horizontal-tab***%I%**

%A% 另一种用于构建非 HP-UX 系统程序文件的 *what*(1) 字符串的简写形式。

%A% = %Z%%Y% %M% %I%%Z%

外部语言环境影响

环境变量

LC_CTYPE 用于确定将文本解释为单字节和（或）多字节字符的方法。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或 **LC_MESSAGES** ，或者将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而不使用 **LANG** 。如果任一国际化变量中包含无效设置，则 **get** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

可使用 *sccshelp*(1) 获得进一步说明。

警告

如果有效用户在包含 SCCS 文件的目录中拥有写入权限（显式或隐式），但是实际用户却没有，则使用 **-e** 选项时可以仅命名一个文件。

当使用 **-i** 选项将更改合并到已删除（可能因为不小心）并在随后重新插入文件的文件部分，将出现意外的结果。

使用 **-g** 时不能生成 *l-file* 。也就是说， **-g -l** 不起作用。

文件

通过 **get** 可创建几个辅助文件。这些文件通常称为 *g-file* 、 *l-file* 、 *p-file* 和 *z-file* 。连字符之前的字母称作标记。辅助文件名是利用 SCCS 文件名形成的：所有 SCCS 文件名的最后一部分必须是 **s.***module-name* 形式，辅助文件通过将前导的 **s** 替换为标记来命名。 *g-file* 是该方案的例外情况： *g-file* 通过删除 **s.** 前缀来命名。例如 **s.xyz.c** ，辅助文件名将分别是 **xyz.c** 、 **l.xyz.c** 、 **p.xyz.c** 和 **z.xyz.c** 。

包含的所生成文本的 *g-file*

在当前目录中创建（除非使用 **-p** 选项）。 *g-file* 在所有情况下创建，无论 **get** 是否生成文本行。它由实际用户拥有。如果使用或暗含 **-k** 选项，其模式为 644；否则，其模式为 444。只有实际用户才需要当前目录中的写入权限。

l-file 包含一个表，它显示在生成检索文本时应用的 delta 版。如果使用 **-l** 选项， *l-file* 将在当前目录中创建；其模式为 444，它由实际用户拥有。只有实际用户才需要当前目录中的写入权限。

l-file 中的行具有以下格式：

1. 如果应用了 delta 版，则为空白字符；
   否则是 ***** 。

2. 如果应用了 delta 版或未应用并忽略了 delta 版，则为空白字符；
   如果未应用且未忽略 delta 版，则为 ***** 。

3. 指示为何应用或未应用 delta 版的“特殊”原因的代码：

   I ：已包含。
   X ：已排除。
   C ：截止（通过 **-c** 选项）。

4. 空白字符。

5. SCCS 标识 (SID)。

6. 制表符。

7. 创建日期和时间（格式为 YY/MM/DD HH:MM:SS）。

8. 空白字符。

9. 创建 *delta* 更改量的用户的登录名。

注释和 MR 数据在后继行上列出，并缩进一个横向制表符的距离。空白行将终止每个条目。

p-file 用于将 **get** 生成的信息随 **-e** 选项传递到 *delta* 更改量。其内容也用于防止以 **-e** 选项在随后为相同的 SID 执行 **get** ，直到执行 *delta* 更改量或者在 SCCS 文件中设置连接编辑标志 **j** （请参阅 *admin*(1) ）。 *p-file* 在包含 SCCS 文件的目录中创建，有效用户必须在该目录中拥有写入权限。其模式是 644，它由有效用户拥有。 *p-file* 的格式为：获取的 SID，后接一个空白字符，后接新 delta 版在创建时具有的 SID，后接一个空白字符，后接实际用户的登录名，后接一个空白字符，后接执行 **get** 的日期和时间，后接一个空白字符和 **-i** 选项参数（如果有），后接一个空白字符和 **-x** 选项参数（如果有），后接一个换行符。 *p-file* 中随时可能有任意个行；任何两行都不能具有相同的新 delta 版 SID。

z-file 用作防止同时更新的 锁定机制。其内容是创建它的命令（即 **get** ）的二进制（2 字节）进程 ID。 *z-file* 在 **get** 执行过程中包含 SCCS 文件的目录中创建。与 *p-file* 相同的保护限制适用于 *z-file* 。 *z-file* 创建为模式 444。

另请参阅

admin(1)、 delta(1)、 prs(1)、 sccshelp(1)、 what(1)、 sccsfile(4)。

符合的标准

get：SVID2、SVID3、XPG2、XPG3、XPG4

名称

getaccess - 列出文件的访问权限

概要

getaccess [**-u** *user*] [**-g** *user*] *group* [**,** *group*] ...] [**-n**] *file* ...

getaccess -r [**-n**] *file* ...

说明

getaccess 为指定文件列出调用方（即有效用户 ID 、有效组 ID 以及补充组列表）的有效访问权限。缺省情况下，该命令将用户访问权限的符号形式输出到命名的文件： **r** 或 **-** （表示读取或不读取）、 **w** 或 **-** （表示写入或不写入）以及 **x** 或 **-** （表示执行或不执行；对于目录为搜索/不搜索），后接文件名。

选项

getaccess 识别下列选项和命令行参数：

-u *user*
列出给定用户而不是调用方的访问权限。 *user* 可以是已知的用户名、有效的 ID 号或表示文件所有者 ID 的 @。如果请求有关多个文件的信息，每个文件的 @ 值可能不同。

该选项仅设置用户 ID 。除非还指定 **-g** ，否则将通过调用方的有效组 ID 和补充组 ID检查访问权。

-g *group* [, *group*] ...]
列出给定组（而非调用方的有效组 ID 和补充组列表）的访问权。 *group* 可以是已知的组名、有效的 ID 号或表示文件组 ID 的 @。如果请求有关多个文件的信息，每个文件的 @ 值可能不同。

-r
使用调用方的实际用户 ID 、组 ID 和补充组列表（而不是有效的 ID 值）列出访问权。

-n
以数字形式列出所请求的每个文件的访问权（八进制位 **0**..**7** ，而不是 **rwx** ）。位值 **R_OK** 、 **W_OK** 和 **X_OK** 在文件 **<unistd.h>** 中定义。

关于使用访问控制列表检查访问权的相关信息，请参阅 *acl*(5) 和 *aclv*(5) 。

此外，对于只读文件系统上的文件或者所执行的共享文本程序，将清除写入位。对于为写入而打开的共享文本程序将不关闭执行位，这是因为无法确定为写入而打开的文件是否是共享文本文件。

具有适当特权的进程具有对所有文件的读写访问权。此外，对于只读文件系统上的文件或者所执行的共享文本程序，将拒绝写入访问权。当且仅当文件不是常规文件或者在文件的任何 ACL 条目中设置了执行位时，才允许执行访问权。

要成功使用 **getaccess** ，调用方必须在 *file* 路径名的每一个目录部分中具有搜索访问权。 **getaccess** 首先通过使用调用方的有效 ID 验证搜索访问权，无论指定了什么用户和组 ID 。这不同于调用方可以搜索路径但检查其访问权的用户不具有该文件访问权的情况。

注意： **-** 的文件名参数对于 **getaccess** 不具有特殊含义（如标准输入）。

外部语言环境影响

环境变量

LANG 用于确定显示消息的语言。

如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省的“C”（请参阅 *lang*(5)）而非 **LANG** 。如果任一国际化变量中包含无效设置，则 **getaccess** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

返回值

getaccess 返回下列值之一：

0 成功完成。

1 **getaccess** 调用错误或者遇到未知的用户名或组名。相应的消息将输出到标准错误。

2 文件不存在或无法访问（由调用方）。 **getaccess** 将相应的消息输出到标准错误，继续执行，然后在完成时返回值 2。

举例

以下命令将调用方的访问权限输出到 *file1* ，所使用的是该文件的组 ID 而不是调用方的有效组 ID 和组列表。

getaccess -g@ file1

下面是检查组 **red** 和 **19** 中的用户 **ggd** 对当前目录中所有文件的访问权的方法，访问权限表示为八进制值。

getaccess -u ggd -g red,19 -n .* *

下面是列出 **mydir** 下所有文件的访问权限的方法。

find mydir -print | sort | xargs getaccess

作者

getaccess 由 HP 开发。

文件

/etc/passwd

/etc/group

另请参阅

chacl(1)、 getacl(1)、 lsacl(1)、 setacl(1)、 getaccess(2)、 glossary(9)。

名称

getacl - 列出文件的访问控制列表 (ACL)（仅限 JFS 文件系统）

概要

/usr/bin/getacl [**-ad**] *file...*

说明

对每一个是常规文件、特殊文件或命名管道的参数， **getacl** 将显示所有者、组和访问控制列表 (ACL)。对每一个目录参数， **getacl** 将显示所有者、组和 ACL 和（或）缺省 ACL。只有目录包含缺省 ACL。

如果指定了 **-a** 选项，则将显示文件的文件名、所有者、组和 ACL。如果指定了 **-d** 选项，则将显示文件的文件名、所有者、组和缺省 ACL （如果有）。如果未指定任何选项，则将显示文件名、所有者、组以及 ACL 和缺省 ACL（如果有）。

可以对不支持 ACL 的文件系统执行该命令。它将根据权限位报告 ACL，其中仅包括拥有用户、拥有组、类和其他条目。

当在命令行上指定多个文件时，将使用空行来分隔每个文件的 ACL。ACL 的格式为:

> **# file:** *filename*
> **# owner:** *uid*
> **# group:** *gid*
> **user::***perm*
> **user:***uid***:***perm*
> **group::***perm*
> **group:***gid***:***perm*
> **class:***perm*
> **other:***perm*
> **default:user::***perm*
> **default:user:***uid***:***perm*
> **default:group::***perm*
> **default:group:***gid***:***perm*
> **default:class:***perm*
> **default:other:***perm*

前三行显示了文件名、文件所有者和文件拥有组。请注意，当仅指定 **-d** 选项并且文件没有缺省 ACL 时，将只显示这三行内容。

没有用户 ID 的 **user** 条目表示将要授予文件所有者的权限。一个或多个附加 **user** 条目表示将要授予指定用户的权限。没有组标识符的 **group** 条目表示将要授予文件拥有组的权限。一个或多个附加 **group** 条目表示将要授予指定组的权限。 **other** 条目表示将要授予其他用户的权限。

default 条目（ default:user 、 **default:group** 和 **default:other** ）仅针对目录存在，它们表示将要添加到创建于目录中的文件的缺省用户、缺省组和缺省其他条目。

uid 是登录名称，如果在系统的口令文件中没有 *uid* 的条目，则是用户 ID； *gid* 是组的名称，如果在系统的组文件中没有 *gid* 的条目，则是组 ID； *perm* 是三个字符的字符串，其组成字母代表单独的自由选定的访问权限： **r** （读）、 **w** （写）、 **x** （执行/搜索），或占位符 **-** 。 *perm* 将以下列顺序显示： **rwx** 。如果 ACL 条目没有授予权限，则将显示占位符。

当执行访问检查时， ACL 条目将按其被评估的顺序显示。某目录上可能已有的缺省 ACL 条目对访问检查没有影响。

文件所有者权限位表示拥有用户 ACL 条目所具有的权限。文件组类权限位表示任何附加用户条目、附加组条目或拥有组条目可能授予的最大权限。文件其他权限位表示其他 ACL 条目具有的权限。如果用户调用 **chmod** 命令并更改文件组类权限位，则由附加 ACL 条目授予的权限可能会受到限制。

为了表示文件组类权限位对 ACL 条目的限制，将在每个受影响的条目之后显示 **getacl** ，其文本格式为 **#effective:***perm*，其中 *perm* 将只显示实际授予的权限。

举例

假定文件为 **filea** ，并且 ACL 有六个条目，则命令

```
$ getacl filea
```

将输出：

```
# file: filea
# owner: fletcher
# group: us
user::rwx
user:spy:---
user:archer:rw-
group::r--
class:rw-
other:---
```

假定文件为 **filea** ，并且 ACL 有六个条目，则在发出命令 **chmod 700 filea** 之后，命令

```
$ getacl filea
```

将输出：

```
# file: filea
# owner: fletcher
# group: us
user::rwx
user:spy:---
user:archer:rw- #effective:---
group::r-- #effective:---
class:---
```

other:---

假定目录为 **fileb** ，并且 ACL 包括缺省条目，则命令

$ getacl -d fileb

将输出：

file: fileb

owner: fletcher

group: us

default:user::rwx

default:user:spy:---

default:group::r--

default:other:---

假定目录为 **fileb** ，则命令

$ getacl fileb

将输出：

file: fileb

owner: fletcher

group: us

user::rwx

user:spy:---

user:archer:rw-

group::r--

other:---

default:user::rwx

default:user:spy:---

default:group::r--

default:other:---

注意事项

getacl 的输出将采用供 **setacl** 命令的输入使用的正确格式。如果将 **getacl** 的输出重定向到一个文件，则该文件可以用作 **setacl** 的输入。使用这种方法，用户可以轻松地将一个文件的 ACL 指定给另一个文件。

文件

/etc/passwd 供用户 ID 使用

/etc/group 供组 ID 使用

另请参阅

acl(2)、 aclsort(3C)、 chmod(1)、 ls(1)、 setacl(1)。

名称

getconf - 获取系统配置值

概要

getconf [**-v** *specification*] *system_var*

getconf [**-v** *specification*] *system_var* *pathname*

说明

getconf 命令提供 *confstr*(3C) 、 *pathconf*(2) 和 *sysconf*(2) 库例行程序和系统调用的接口。

system_var 参数指定 **confstr()** 、 **pathconf()** 或 **sysconf()** 中所需的配置值。将第一种概要形式用于涉及 **confstr()** 或 **sysconf()** 的查询（在下面的第一个表中）。将第二种概要形式用于涉及 **pathconf()** 的查询（在下面的第二个表中）。对于涉及 **pathconf()** 的查询，应该指定 *pathname* 操作数。

选项

getconf 识别以下选项：

-v *specification* 返回与 HP-UX 支持的特定编译环境对应的配置变量。如果未指定 **-v** 选项，则 *specification* 缺省为 XBS5_ILP32_OFF32。有关可能的规范和含义，请参阅下表。

规范	**int**	**long**	**pointer**	**off_t**
XBS5_ILP32_OFF32	32	32	32	32
XBS5_ILP32_OFFBIG	32	32	32	>=64
XBS5_LP64_OFF64	32	64	64	64
XBS5_LPBIG_OFFBIG	>=32	>=64	>=64	>=64

外部语言环境影响

环境变量

LC_MESSAGES 用于确定显示消息的语言。

如果在环境中未指定 **LC_MESSAGES** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 。

如果任一国际化变量中包含无效设置，则 **getconf** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

getconf 返回的错误代码是：

- **0** 成功。返回了对应于操作数的值。
- **1** 缺少一个或多个操作数。
- **2** 无法识别操作数。
- **3** *pathname* 无效或无法访问。

举例

请求每秒间隔数:

getconf CLK_TCK

请求文件链接数的最大值:

getconf LINK_MAX /etc/passwd

询问该实现是否支持多个位置域:

getconf CCNUMA_SUPPORT

getconf 提供以下输出:

1	如果该实现支持多个位置域。
undefined	如果该实现不支持多个位置域。

支持的其他查询包括:

ARG_MAX	*_BC_BASE_MAX*	*BC_DIM_MAX*
BS_SCALE_MAX	*BC_STRING_MAX*	*CHARCLASS_NAME_MAX*
CHAR_BIT	*CHAR_MAX*	*CHAR_MIN*
CHILD_MAX	*CLK_TCK*	*COLL_WEIGHTS_MAX*
CPU_CHIP_TYPE	*CS_MACHINE_IDENT*	*CS_PARTITION_IDENT*
CS_PATH	*CS_MACHINE_SERIAL*	*EXPR_NEST_MAX*
HW_CPU_SUPP_BITS	*HW_32_64_CAPABLE*	*INT_MAX*
INT_MIN	*IPMI_INTERFACE*	*KERNEL_BITS*
LINE_MAX	*LONG_BIT*	*LONG_MAX*
LONG_MIN	*MACHINE_IDENT*	*MACHINE_MODEL*
MACHINE_SERIAL	*MB_LEN_MAX*	*NGROUPS_MAX*
NL_ARGMAX	*NL_LANGMAX*	*NL_MSGMAX*
NL_NMAX	*NL_SETMAX*	*NL_TEXTMAX*
NZERO	*OPEN_MAX*	*PARTITION_IDENT*
PATH	*_POSIX_ARG_MAX*	*_POSIX_JOB_CONTROL*
_POSIX_NGROUPS_MAX	*_POSIX_OPEN_MAX*	*_POSIX_SAVED_IDS*
_POSIX_SSIZE_MAX	*_POSIX_STREAM_MAX*	*_POSIX_TZNAME_MAX*
_POSIX_VERSION	*POSIX_ARG_MAX*	*POSIX_CHILD_MAX*
POSIX_JOB_CONTROL	*POSIX_LINK_MAX*	*POSIX_MAX_CANON*
POSIX_MAX_INPUT	*POSIX_NAME_MAX*	*POSIX_NGROUPS_MA*
POSIX_OPEN_MAX	*POSIX_PATH_MAX*	*POSIX_PIPE_BUF*
POSIX_SAVED_IDS	*POSIX_SSIZE_MAX*	*POSIX_STREAM_MAX*

POSIX_TZNAME_MAX	***POSIX_VERSION***	***POSIX2_BC_BASE_MAX***
POSIX2_BC_DIM_MAX	***POSIX2_BC_SCALE_MAX***	***POSIX2_BC_STRING_MAX***
POSIX2_C_BIND	***POSIX2_C_DEV***	***POSIX2_C_VERSION***
POSIX2_CHAR_TERM	***POSIX_CHILD_MAX***	***POSIX2_COLL_WEIGHTS_MAX***
POSIX2_EXPR_NEST_MAX	***POSIX2_FORT_DEV***	***POSIX2_FORT_RUN***
POSIX2_LINE_MAX	***POSIX2_LOCALEDEF***	***POSIX2_RE_DUP_MAX***
POSIX2_SW_DE	***POSIX2_UPE***	***POSIX2_VERSION***
PSET_SUPPORT	***SC_PASS_MAX***	***SC_XOPEN_VERSION***
SCHAR_MAX	***SCHAR_MIN***	***SHRT_MAX***
SHRT_MIN	***SSIZE_MAX***	***STREAM_MAX***
RE_DUP_MAX	***TMP_MAX***	***TZNAME_MAX***
UCHAR_MAX	***UINT_MAX***	***ULONG_MAX***
USHRT_MAX	***WORD_BIT***	***XOPEN_VERSION***
XOPEN_XCU_VERSION	***XOPEN_XPG2***	***XOPEN_XPG3***
XOPEN_XPG4		

XBS5_ILP32_OFF32_CFLAGS	***XBS5_ILP32_OFF32_LDFLAGS***
XBS5_ILP32_OFF32_LIBS	***XBS5_ILP32_OFF32_LINTFLAGS***
XBS5_ILP32_OFFBIG_CFLAGS	***XBS5_ILP32_OFFBIG_LDFLAGS***
XBS5_ILP32_OFFBIG_LIBS	***XBS5_ILP32_OFFBIG_LINTFLAGS***
XBS5_LP64_OFF64_CFLAGS	***XBS5_LP64_OFF64_LDFLAGS***
XBS5_LP64_OFF64_LIBS	***XBS5_LP64_OFF64_LINTFLAGS***

支持的要求第二个参数的查询包括：

LINK_MAX	***MAX_CANON***	***MAX_INPUT***
NAME_MAX	***PATH_MAX***	***PIPE_BUF***
_POSIX_CHOWN_RESTRICTED	***_POSIX_LINK_MAX***	***_POSIX_MAX_CANON***
_POSIX_MAX_INPUT	***_POSIX_NO_TRUNC***	***_POSIX_NAME_MAX***
_POSIX_PATH_MAX	***_POSIX_PIPE_BUF***	***_POSIX_VDISABLE***
POSIX_CHOWN_RESTRICTED	***POSIX_NO_TRUNC***	***POSIX_VDISABLE***

注释

有关 **getconf** 支持的参数的完整列表，请参阅 **sysconf()** 、 **pathconf()** 和 **confstr()** 的手册页。

作者

getconf 由 HP 和 POSIX 开发。

另请参阅

pathconf(2)、 sysconf(2)、 confstr(3C)。

符合的标准

getconf：POSIX.2、XPG4

名称

getopt - 分析命令选项

概要

getopt *optstring args*

说明

getopt 用于分解命令行中的选项，以便于 Shell 过程进行分析以及检查选项是否合法。 *optstring* 是可识别的选项字母组成的字符串（请参阅 *getopt*(3C)）。如果某字母后紧接一个冒号，那么该选项应该具有一个可能由空格（也可能没有空格）与该选项分隔的参数。

Shell 的位置参数 ($1 $2 ...) 将被重置，以便在每个选项有一个前导 **-**，并且每个选项处于其自己的位置参数所指定的位置上；每个选项参数还将被分析为自己的位置参数。

getopt 通过识别两个连字符 (**--**) 来确定选项已处理完毕。如果没有两个连字符， **getopt** 会将 **--** 放置在选项列表的末尾。

getopt 最常用于 Shell 的 **set** 命令中（请参阅下面的示例），此时 **getopt** 将命令行转换为更易于分析的形式。**getopt** 将修改后的命令行写入到标准输出。

外部语言环境影响

环境变量

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_MESSAGES** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。

如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5)）而非 **LANG**。如果任一国际化变量包含无效设置，则 **getopt** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5)。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

当 **getopt** 遇到了 *optstring* 中不包含的选项字母时，会在标准错误中输出一条错误消息。

举例

以下代码段处理一个命令的参数，该命令可以带有选项 **a** 或 **b** 以及选项 **o**（需要一个参数）：

```
set -- `getopt abo: $*`

if [ $? -ne 0 ]; then
   echo $USAGE
   exit 2
fi

while [ $# -gt 0 ]; do
```

```
    case $1 in
    -a | -b)
        FLAG=$1
        shift
        ;;
    -o)
        OARG=$2
        shift 2
        ;;
    --)
        shift
        break
        ;;
    esac
done
```

以上代码与以下任一命令等效:

```
cmd -aoarg file file
cmd -a -o arg file file
cmd -oarg -a file file
cmd -a -oarg -- file file
```

警告

getopt 选项参数不得是空字符串，也不得包含嵌入的空白字符。

另请参阅

sh(1)、 getopt(3C)。

名称

getopts - 解析实用程序（命令）选项

概要

getopts *optstring name* [*arg* ...]

说明

getopts 用于从参数列表中检索选项和选项参数。

每次调用 **getopts** 时，该命令都会将下一个选项的值放置到由 **name** 操作数所指定的 Shell 变量中，并将要处理的下一个参数的索引放置到 Shell 变量 **OPTIND** 中。只要调用了 Shell， **OPTIND** 就会被初始化为 1。

当选项需要选项参数时， **getopts** 就会将该选项参数放置到 Shell 变量 **OPTARG** 中。如果没有找到选项，或找到的选项不具有选项参数，则将取消设置 **OPTARG** 。

如果在期望有选项字符时，却发现在 *optstring* 操作数中没有包含选项字符，则由 *name* 指定的 Shell 变量将被设置为问号 (**?**) 字符。在这种情况下，如果 *optstring* 中的第一个字符是冒号 (**:**) ，则 Shell 变量 **OPTARG** 将被设置为找到的选项字符，但是不将输出写到标准错误中；否则，将取消设置 Shell 变量 **OPTARG** ，并将诊断消息写到标准错误中。可以将该情况视为以调用应用程序时显示参数的方式所检测到的错误，而不是 **getopts** 处理中的错误。

如果选项参数缺失：

- 如果 *optstring* 的第一个字符是一个冒号，则由 *name* 所指定的 Shell 变量将被设置为冒号，而 Shell 变量 **OPTARG** 将被设置为所找到的选项字符。
- 否则，由 *name* 所指定的 Shell 变量将被设置为问号，而 Shell 变量 **OPTARG** 将被取消设置，同时将诊断消息写到标准错误中。可以将该情况视为以调用应用程序时显示参数的方式所检测到的错误，而不是 **getopts** 处理中的错误；诊断消息将按照规定来写，但是退出状态为零。

遇到选项末尾时， **getopts** 会退出，其返回值大于零。 Shell 变量 **OPTIND** 被设置为第一个非选项参数的索引，如果在第一个 **--** 参数之前没有其他的非选项参数的话，就将该参数视为选项参数，或者如果没有任何非选项参数的话，就将该第一个参数视为值 **$#** + 1； *name* 变量被设置为问号字符。以下任何一种情况都可以识别选项的末尾：特殊选项 **--** ，发现不以 **-** 开头的参数或遇到一个错误。

Shell 变量 **OPTIND** 和 **OPTARG** 对于 **getopts** 的调用者来说是本地变量，而且在缺省情况下不导出。

由 *name* 操作数指定的 Shell 变量 **OPTIND** 和 **OPTARG** 会影响当前 Shell 的执行环境。

操作数

支持下列操作数：

optstring 指包含由调用 **getopts** 的实用程序所识别的选项字符的字符串。如果字符后面是冒号 (**:**)，则选项应含有一个参数，而且该参数应是一个独立参数。应用程序应指定选项字符及其选项参数作为独立参数，但是无论是否进行了指定， **getopts** 都会将需要参数的选项字符之后的字符解释为参数。如果调用 **getopts** 时，明确为空的选项参数不是一个独立参数，则此选项参数不需要被识别。在应用程序中，禁止将字符问号 (**?**) 和冒号 (**:**) 作

为选项字符来使用。使用其他非字母数字的选项字符会产生未指明的结果。如果从选项字符中找不到可以作为独立参数的选项参数，则 **OPTARG** 中的值将被从选项字符和 **-** 中去除。如果选项字符是未知的或选项参数缺失，则 *optstring* 中的第一个字符将决定 **getopts** 的行为方式。

name 指被 **getopts** 设置为所找到的选项字符的 Shell 变量的名称。

getopts 在缺省情况下会解析被传送到调用 Shell 程序的位置参数。如果已经给定 *args* ，则将解析这些参数，而不是解析位置参数。

外部语言环境影响

环境变量

下列环境变量将影响 **getopts** 实用程序的执行：

OPTIND 由 **getopts** 作为要处理的下一个参数的索引来使用。

错误

只要检测到错误，并且 *optstring* 操作数的第一个字符不是冒号 (**:**) ，就会将诊断消息写到标准错误中，其中包含以未指定的格式输出的下列信息：

- 该消息将识别调用程序的名称。在调用 **getopts** 实用程序时，调用程序的名称将会是 Shell 特殊参数 0 的值。名称等同于：

  basename "$0"

  可以被使用。

- 如果发现一个未在 *optstring* 中指定的选项，则将识别该错误，并将在消息中识别此无效选项字符。

- 如果发现一个需要具有选项参数但却没有选项参数的选项，则将识别该错误，并将在消息中识别此无效选项字符。

举例

因为 **getopts** 会影响当前 Shell 的执行环境，所以通常情况下，会将其作为一个 Shell 的常规内置命令。如果在 subshell 或独立的实用程序执行环境中（例如以下环境之一）调用该命令：

(getopts abc value "$@")
nohup getopts ...
find -exec getopts ...\;

则它不会影响调用者环境中的 Shell 变量。

请注意，即使已更改位置参数，Shell 函数也会将 **OPTIND** 与正在调用的 Shell 共享。使用 **getopts** 来解析其参数的函数应在进入函数时保存 **OPTIND** 的值，并在返回前恢复该值。但是也有这样的情况，即函数因正在调用的 Shell 而必须更改 **OPTIND** 。

下面的示例脚本可解析并显示其参数：

aflag=

```
bflag=

while getopts ab: name
  do
    case $name in
    a)
        aflag=1;;
    b)
        bflag=1
        bval="$OPTARG";;
    ?)
        printf "Usage: %s: [-a] [-b value] args\n" $0
        exit 2;;
    esac
  done
if [ ! -z "$aflag" ] ; then
    printf "Option -a specified\n"
fi
if [ ! -z "$bflag" ] ; then
    printf "Option -b "%s" specified\n" "$bval"
fi
shift $(($OPTIND -1))
printf "Remaining arguments are: %s\n" "$*"
```

另请参阅

getopt(1)、 ksh(1)、 sh-posix(1)、 sh(1)、 getopt(3C)。

符合的标准

getopts：XPG4、POSIX.2

名称

getprivgrp - 获取组的特殊属性

概要

getprivgrp [**-g**| *group_name*]

说明

getprivgrp 列出由 **setprivgrp** 设置的特权组访问权限（请参阅 *setprivgrp*(1M)）。如果提供了 *group_name* ，则仅列出该组的访问权限。如果调用方不是 *group_name* 的成员，则不显示任何信息。如果使用了 **-g** ， **getprivgrp** 将列出已授予所有组的访问权限。否则，将列出调用方所属的所有特权组的访问权限。

超级用户是所有组的成员。访问权限包括 **RTPRIO** 、 **RTSCHED** 、 **MLOCK** 、 **CHOWN** 、 **LOCKRDONLY** 、 **SETRUGID** 、 **FSSTHREAD** 、 **SPUCTL** 、 **PSET** 、 **MPCTL** 和 **SERIALIZE** 。有关这些权限功能的说明，请参阅 *setprivgrp*(1M) 。

作者

getprivgrp 由 HP 开发。

另请参阅

setprivgrp(1M)、 getprivgrp(2)、 privgrp(4)。

名称

gprof - 显示调用图形配置文件数据

概要

gprof [*options*] [*a.out* [*gmon.out ...*]]

说明

gprof 命令生成 C++、 C 和 FORTRAN 程序的执行配置文件。所调用例程的结果合并到每个调用方的配置文件中。配置文件数据从调用图形配置文件（缺省为 gmon.out ）中提取，该配置文件由使用 **aCC** 、 **cc** 和 **f90** 的 **-G** 选项编辑的程序所创建。 **-G** 选项还会链接为配置处理而编译的库例程版本。

在基于 Itanium(R) 的系统上， **gprof** 支持多共享库配置处理。在 PA-RISC 系统上， **gprof** 支持单共享库配置处理。有关详细信息，请参阅下面的共享库配置处理部分。

读取正在进行配置处理的加载模块的符号表，并将其与调用图形配置文件 (**gmon.out**) 相关联。要具有完整的调用图形，不得剪切任何加载模块符号表；即，任何编译均不得使用 **-x** 选项。如果指定了多个配置文件， **gprof** 输出将在给定的配置文件中显示配置文件信息的摘要。

首先，给定平面配置文件，它类似于 **prof** 提供的配置文件（请参阅 *prof*(1) ）。该列表提供加载模块中每个函数的总执行时间和调用计数，它们按时间降序排列。在基于 Itanium 的系统上，还将报告每个函数的模块索引，指示在其中定义该函数的加载模块。

下一步，将这些时间沿着调用图形的边缘进行扩展。 **gprof** 会显示调用图形中的所有周期。周期中的所有调用共享该周期的时间。第二个列表显示按照所表示时间排序的函数，其中包括其调用图形派生对象的时间。每个函数条目之下将显示其（直接）调用图形子项，以及它们的时间扩展到该函数的方式。函数之上的类似显示指示该函数的时间和其派生对象的时间如何扩展到其（直接）调用图形父项。

另外还将显示周期，以周期条目作为一个整体，而周期成员的列表中包括每个成员在总时间中所占的比例以及该周期的调用计数。

在基于 Itanium 的系统上，最终将提供所有模块索引到模块名的映射。未进程配置处理的模块将在输出的顶部报告。

共享库配置处理

32 位和 64 位基于 Itanium 的系统支持共享库的 **gprof** 配置处理。在 PA-RISC 系统上，仅支持 32 位共享库配置处理。

基于 **Itanium** 的系统上

环境变量 **LD_PROFILE** 确定哪些加载模块进行配置处理。设置 **LD_PROFILE=ALL** 将对所有加载模块进行配置处理；也就是说，报告所有可加载模块（包括 **a.out** ）的时间和调用计数信息。如果设置 **LD_PROFILE=***ldm1***:***ldm2* ，则仅对可加载模块 *ldm1* 和 *ldm2* 进行配置处理。 *ldm1* 和 *ldm2* 不是完整路径名；它们是在可执行文件中记录的名称，这些名称可以使用 *chatr*(1) 来显示。如果未设置 **LD_PROFILE** ， **gprof** 将假定 **LD_PROFILE=ALL** 。

环境变量 **LD_PROFILEBUCKET_SIZE** 控制配置处理计数器的大小。该变量的可接受值是 16 或 32。计数器大小也可在编译时使用 **+profilebucketsize** 选项指定。运行时值将覆盖编译时值。如果计数器大小设置为 16 或 32 之外的值，则将发出警告；此时将使用在编译时指定的值。计数器的缺省值是 16，如果未指定有效值，将使用该缺省值。有关详细信息，请参阅 *cc*(1) 中 **+profilebucketsize** 选项的说明。

在程序终止时， **gprof** 将逐个模块地转储 **gmon.out** 中的所有配置处理信息， **gprof** 命令将读取该信息，并将其与加载模块中的相应函数进行匹配。

PA-RISC 系统上

要对共享库进行配置处理，请将 **LD_PROFILE** 设置为要进行配置处理的共享库的路径。（有关详细信息，请参阅《HP-UX Linker and Libraries User's Guide》）。不要使用 **-G** 选项来编译程序以进行共享库配置处理。不要链接可执行文件 **gcrt0.o** 或 **mcrt0.o** 。它将打开 **a.out** 的配置处理，该配置处理与共享库的配置处理不兼容。可以对可执行文件或共享库中任一种进行配置处理，但不能对这两者同时进行配置处理。

将 **LD_PROFILE** 设置为用来调用 **shl_load** 的确切字符串。如果库是隐式加载的， **LD_PROFILE** 必须匹配 **a.out** 中编码的路径。通过对可执行文件运行 **ldd** 命令，可以找到该值。

在程序终止时，运行时库将生成其前面加有共享库名的配置文件。要获取完整的列表，请提供带共享库名的 **gprof** 命令以及共享库的配置文件作为参数。

选项

gprof 命令可识别下列选项：

-a 禁止输出静态声明的函数。如果给定该选项，有关静态函数的所有相关信息（如时间样例、对其他函数的调用和从其他函数中的调用）将属于在 **a.out** 文件中的静态函数前加载的函数。

-b 禁止输出配置文件中每个字段的说明。

-e *name* 禁止输出例程 *name* 及其所有派生对象的图形配置文件条目（除非它们具有其他未禁止的祖先对象）。可以指定多个 **-e** 选项。对于每个 **-e** 选项，只能指定一个 *name* 。

-E *name* 像上文的 **-e** 一样禁止输出例程 *name* （及其派生对象）的图形配置文件条目，并且还从总时间和时间百分比计算中排除在 *name* （及其派生对象）中所用的时间。 **-E mcount -E mcleanup** 是缺省值。

-f *name* 仅输出指定例程 *name* 及其派生对象的图形配置文件条目。可以指定多个 **-f** 选项。对于每个 **-f** 选项，只能指定一个 *name* 。

-F *name* 仅输出例程 *name* 及其派生对象的图形配置文件条目（与上面的 **-f** 相同），并且还仅使用输出例程的时间计算总时间和时间百分比。可以指定多个 **-F** 选项。对于每个 **-F** 选项，只能指定一个 *name* 。 **-F** 选项将覆盖 **-E** 选项。

-p 仅生成平面配置文件输出，该输出与 **prof** 提供的输出完全类似（请参阅 *prof*(1)）。

-s 生成配置文件 **gmon.sum** ，它显示所有指定的配置文件中的配置文件信息的摘要。该摘要配置文件可以提供给 **gprof** 的后继执行（可能还带有 **-s** 选项），以累计 **a.out** 文件多次运行的配置文件数据。对于所有运行， **LD_PROFILE** 都应设置为相同的字符串。

-t 仅生成 **gprof** 中的静态输出。它用于进行测试。它将从常规 **gprof** 输出去除所有时间信息，而只报告调用计数部分。

-z 显示利用率为零的例程（由调用计数和累计时间来指示）。

环境变量 **GPROFDIR** 控制经配置处理的程序所创建的文件的名称。如果未设置 **GPROFDIR** ，则程序终止时，将在当前目录中生成 **gmon.out** 。如果 **GPROFDIR=***string* ，则将生成 *string***/***pid.progname* ，其中 *progname* 是删除任何路径前缀的 **argv[0]** ， *pid* 是该程序的进程 ID。如果 **GPROFDIR** 设置为空字符串，则不生成配置处理输出。

外部语言环境影响

环境变量

LD_PROFILE 确定要进行配置处理的模块。

LD_PROFILEBUCKET_SIZE
控制配置处理计数器的大小。

GPROFDIR 控制经配置处理的程序所创建的文件的路径和名称。

示例

对基于 Itanium 的系统上的 **a.out** 和 **libtest.so** 进行配置处理:

```
$ cat > test.c
void a()
{
    printf("I in a\n");
}

$ cc -c +Z -G test.c
$ ld -b -o libtest.so.1 test.o
$ ln -s ./libtest.so.1 libtest.so
$ cat > main.c
extern void a();
main()
{
    printf("Hello world\n");
    a();
}

$ cc -G main.c  -L. -ltest
```

```
$ export LD_PROFILE=a.out:libtest.so
$ export LD_PROFILEBUCKET_SIZE=16
$ ./a.out
hello world
I in a

$ unset LD_PROFILE
$ unset LD_PROFILEBUCKET_SIZE
$ ls gmon.out
gmon.out

$ gprof
```

对 PA-RISC 系统上的 **libc.sl** 进行配置处理:

```
$ cat > test.c
main()
{
    printf("hello world\n");
}

$ cc test.c -lc
$ ldd a.out
    /usr/lib/libc.2 =>      /usr/lib/libc.2
    /usr/lib/libdld.2 =>    /usr/lib/libdld.2
    /usr/lib/libc.2 =>      /usr/lib/libc.2

$ export LD_PROFILE=/usr/lib/libc.2
$ ./a.out
hello world

$ unset LD_PROFILE
$ ls libc.2.profile
libc.2.profile

$ gprof /usr/lib/libc.2 libc.2.profile
```

警告

请留意量化误差。虽然显示了抽样的粒度，但抽样粒度在最理想的情况下仍然是统计信息。假定函数的每次执行时间可以表示为该函数的总时间除以调用该函数的次数。因此，沿着调用图形弧线扩展到该函数父项的时间与遍历该弧线的次数直接成比例。

未进行配置处理的父项包含扩展到它们的经配置处理的子项的时间，但它们在调用图形列表中看起来像是自发调用，并且它们的时间不会进一步扩展。同样，信号捕获程序虽然会进行配置处理，看起来也像是自发的（虽然是出于更为不明显的原因）。信号捕获程序的任何配置处理子项应该正确地扩展其时间，但在执行配置处理例程的过程中调用了信号捕获程序时除外，后一种情况下将失去所有数据。

PA-RISC 系统上的 **gprof** 共享库配置处理存在以下限制：

- 不会对本地、静态和隐藏函数进行配置处理。
- 不会对使用 **-B symbolic** 构建的共享库进行配置处理。
- 不收集从库初始设置中进行的任何函数调用。

相关内容

gprof 不能与动态链接的可执行文件（在 HP-UX 10.20 之前的发布版中用 **ld -A** 构建）一起使用。

作者

gprof 由加州大学伯克利分校开发。

文件

a.out*	缺省对象文件。
gmon.out*	缺省的动态调用图形和配置文件。
gmon.sum*	汇总的动态调用图形和配置文件。
/usr/lib/gprof.callg*	调用图形说明。
/usr/lib/gprof.flat*	平面配置文件说明。
/usr/lib/hpux32/libgprof.so	**gprof** 基于 Itanium 的系统上的 32 位共享库
/usr/lib/hpux64/libgprof.so	**gprof** 基于 Itanium 的系统上的 64 位共享库
/usr/lib/libgprof32.sl	**gprof** PA-RISC 系统上的 32 位共享库。
/usr/lib/pa20_64/libgprof.sl	**gprof** PA-RISC 系统上的 64 位共享库。

另请参阅

aCC(1)、 cc(1)、 cc_bundled(1)、 f90(1)、 ld(1)、 prof(1)、 exit(2)、 profil(2)、 sprofil(2)、 monitor(3C)、 smonitor(3C)、 crt0(3)。

《gprof：A Call Graph Execution Profiler》；；Graham S.L.，Kessler P.B.，McKusick M.K.

《Proceedings of the SIGPLAN ’82 Symposium on Compiler Construction》；SIGPLAN Notices；第 17 卷，第 6 期，第 120-126 页，1982 年 6 月。

《HP-UX Linker and Libraries Online User’s Guide》（请参阅 **ld +help** 选项）。

名称

grep、egrep、fgrep - 在文件中搜索模式

概要

通过模式的普通调用

grep [**-E** | **-F**] [**-c**|**-l**|**-q**] [**-bhinsvwx**] *pattern* [*file* ...]

通过（多个）–e 模式调用

grep [**-E** | **-F**] [**-c**|**-l**|**-q**] [**-bhinsvwx**] **-e** *pattern* ... [**-e** *pattern*] ... [*file* ...]

通过 –f 文件调用

grep [**-E** | **-F**] [**-c**|**-l**|**-q**] [**-bhinsvwx**] [**-f** *pattern_file*] [*file* ...]

过时形式:

egrep [**-cefilnsv**] [*expression*] [*file* ...]

fgrep [**-cefilnsvx**] [*strings*] [*file* ...]

说明

grep 命令在输入文本 *files* （缺省为标准输入）中搜索匹配模式的行。通常，找到的每个行将复制到标准输出。**grep** 支持基本正则表达式语法（请参阅 *regexp*(5) ）。 **-E** 选项 (**egrep**) 支持扩展的正则表达式 (ERE) 语法（请参阅 *regexp*(5) ）。 **-F** 选项 (**fgrep**) 使用快速 Boyer-Moore 字符串搜索算法搜索固定的 *strings* 。 **-E** 和 **-F** 选项将 *pattern* 中嵌入的换行符当作替换字符。空表达式或字符串将匹配每一行。

为了向后兼容，保留了 **egrep** 和 **fgrep** 形式。为了实现可移植性，建议使用 **-E** 和 **-F** 选项。

选项

-E 扩展的正则表达式。所指定的每个模式都是一个或多个 ERE 的序列。 ERE 可以用换行符分隔，或者在单独的 **-e** *expression* 选项中提供。如果序列中的任何 ERE 匹配输入行的内容（不包括其尾随换行符），模式就匹配该输入行。使用 **egrep** 可获取相同的功能。

-F 固定的字符串。所指定的每个模式都是一个或多个字符串的序列。字符串可以用换行符分隔，也可在单独的 **-e** *expression* 选项中提供。如果输入行包含序列中的任何字符串，则模式就匹配该输入行。使用 **fgrep** 可获取相同的功能。

-b 每一行之前加有该行所在块的编号。它可用于通过环境定位磁盘块号。块号的计算方法是将从文件中读取的字符数除以 512，然后将结果向下舍入。

-c 仅输出匹配行的计数。

-e *expression* 与简单的 *expression* 参数相同，但用于 *expression* 以连字符 (**-**) 开头的情况。可以使用多个 **-e** 选项来修改多个模式；如果输入行匹配任何指定的模式，则选择该输入行。

-f *pattern_file* 正则 *expression* （**grep** 和 **grep -E** ）或 *strings* 列表 (**grep -F**) 从 *pattern_file* 中提取。

-h 禁止在搜索多个文件时输出文件名。

-i 在比较时忽略大小写的区别。

-l 仅列出包含匹配行的文件的名称（仅列出一次），名称之间用换行符分隔。如果搜索标准输入，在 POSIX 语言环境中将写入 **(standard input)** 的路径名。在其他语言环境中， **(standard input)** 可能会替换为这些语言环境中更合适的内容。

-n 每一行之前加有它在文件中的相对行号（以 1 开始编号）。对于搜索的每个文件，将重置行号。如果指定了 **-c** 、 **-b** 、 **-l** 或 **-q** ，则忽略该选项。

-q （静默）无论是否有匹配行，不将任何内容写入标准输出。在找到第一个匹配行时以零状态退出。覆盖任何将生成输出的选项。

-s 禁止为不存在或不可读的文件生成错误消息。

-v 输出除匹配行之外的所有行。

-w 仅选择包含整词匹配的行。测试条件是匹配字符串必须位于行的开头，或者前加非单词组成字符。同样，它必须位于行尾，或者后接非单词组成字符。单词组成字符包括字母、数字和下划线。

-x （精确）只有当整个输入行匹配固定字符串或正则表达式时，才识别匹配。

除非指定 -h 选项，如果存在多个输入文件，只要生成输出，就会输出文件名。由于字符 **$** 、 ***** 、 **[** 、 **^** 、 **|** 、 **(** 、 **)** 和 **** 对于 Shell 具有特殊含义，因此在 *expression* 中应慎用这些字符。最安全的方法是将整个 *expression* 参数包含在单引号 **('...')** 内。

外部语言环境影响

环境变量

LANG 可确定当 **LC_ALL** 和相应的环境变量（以 **LC_** 开头）都未指定语言环境时，语言环境类别将使用的语言环境。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值 **C** （请参阅 *lang*(5) ）。

LC_ALL 确定用于覆盖各语言环境类别的任何值（通过设置 **LANG** 或以 **LC_** 开头的任何环境变量来指定）的语言环境。

LC_COLLATE 确定在计算正则表达式时使用的排序顺序。

LC_CTYPE 确定文本作为单字节和（或）多字节字符的解释、字符作为字母的分类、 **-i** 选项的大小写信息以及在常规表达式中按字符类表达式匹配的字符。

LC_MESSAGES 用于确定显示消息的语言。

如果任一国际化变量包含无效设置，则命令就会认为所有国际化变量都设置为 **C** 。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

完成后， **grep** 将返回下列值之一:

- **0** 找到一个或多个匹配项。
- **1** 未找到匹配项。
- **2** 语法错误或文件不可访问（即使找到匹配项）。

举例

在 POSIX Shell (*sh*(1)) 中，以下示例通过查找所有包含四个字符串中任意字符串的实例的行来搜索两个文件。

grep -F 'if
then
else
fi' *file1 file2*

请注意，单引号是必需的，它指示 **grep -F** 字符串何时结束，文件名何时开始。

对于 C Shell（请参阅 *csh*(1) ），可使用以下命令:

grep -F 'if\ then\ else\ fi' *file1 file2*

搜索包含以下条目的文件 **address** :

Ken 112 Warring St. Apt. A
Judy 387 Bowditch Apt. 12
Ann 429 Sixth St.

命令:

grep Judy address

输出:

Judy 387 Bowditch Apt. 12

要搜索包含 **Dec** 或 **Nov** 的行的文件，请使用下列命令之一:

grep -E '[Dd]ec|[Nn]ov' *file*
egrep -i 'dec|nov' *file*

在当前目录内的所有文件中搜索 **xyz** :

grep xyz *

在当前目录子树内的所有文件中搜索字符串 **xyz** ，并确保不会因为文件扩展名超过系统参数列表限制而出错:

find . -type f -print |xargs grep xyz

上例不输出包含字符串 **xyz** 的文件的名称。要强制 **grep** 输出文件名，请将另一个参数添加到命令行的 **grep** 命令部分:

find . -type f -print |xargs grep xyz /dev/null

该形式中，第一个文件名是 **find** 生成的名称，第二个文件名是空文件。

警告

（仅适用于 XPG4。）如果指定了 **-q** 选项，并选择了输入行，退出状态将是零（即使检测到错误）。否则，将执行缺省操作。

如果 **-w** 选项是用非单词组成字符指定的，则输出结果将无法预计。

另请参阅

sed(1)、 sh(1)、 regcomp(3C)、 environ(5)、 lang(5)、 regexp(5)。

符合的标准

grep： SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

egrep： SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

fgrep： SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

groups - 显示组成员关系

概要

groups [**-p**] [**-g**] [**-l**] [*user*]

说明

groups 显示了调用方或可选指定 *user* 所属的组。如果调用时未指定任何参数， **groups** 将输出由 **getgroups()** （请参阅 *getgroups*(2) ）返回的当前访问列表。

每个用户都属于口令文件 **/etc/passwd** 指定的组，并可能属于 **/etc/group** 和 **/etc/logingroup** 文件指定的其他组。登录时，用户将获得 **/etc/passwd** 和 **/etc/logingroup** 中指定的组的权限。 **/etc/group** 中指定的组的权限通常仅在使用 **newgrp** （请参阅 *newgrp*(1) ）时可用。如果指定了用户名但未指定任何选项，则 **groups** 将输出所有这些组。

-p 、 **-g** 和 **-l** 选项将输出列表限制为分别在 **/etc/passwd** 、 **/etc/group** 和 **/etc/logingroup** 中指定的那些组。如果未用任一选项指定用户名，则会调用 **cuserid()** 来确定缺省用户名（请参阅 *cuserid*(3S) ）。

组的输出列表将以升序进行排序（请参阅下面的“环境变量”）。

外部语言环境影响

环境变量

LC_COLLATE 用于确定对输出进行排序的顺序。

如果未在环境中指定 **LC_COLLATE** 或将其设置为空字符串，则使用 **LANG** 作为缺省值。如果未设置 **LANG** 或将其设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG**。如果任一国际化变量包含无效设置，则 **groups** 就会认为所有国际化变量都设置为“C”（请参阅 *environ*(5) ）。

举例

检查文件 **/etc/logingroup** 并显示用户 **tim** 所属的所有组：

groups -l tim

作者

groups 由加州大学伯克利分校开发。

文件

/etc/group
/etc/logingroup
/etc/passwd

另请参阅

id(1)、 newgrp(1)、 getgroups(2)、 initgroups(3C)、 cuserid(3S)、 group(4)。

名称

head - 输出前几行

概要

head [**-c**|**-l**] [**-n** *count*] [*file* ...]

过时形式:

head [**-***count*] [*file* ...]

说明

head 在标准输出中输出每个指定文件或标准输入的由 *count* 指定的前几行。如果省略了 *count* ，则缺省为 10。

如果指定了多个 *file* ， **head** 将在每个文件之前输出一行:

==> *file* **<==**

选项

-c 以字节为单位计量输出量。

-*count* 输出单元的数量。该选项是为了实现向后兼容而提供的（请参阅下面的 **-n** ），它与其他所有选项相互排斥。

-l 以行为单位计量输出量；这是缺省设置。

-n *count* 行数（缺省设置）或字节输出量。 *count* 是一个无符号十进制整数。如果未指定 **-n** （或 **-***count*），则缺省数量为 10。该选项提供了与 **-***count* 选项相同的功能，但更为标准。如果系统间的可移植性很重要，建议使用 **-n** 选项。

外部语言环境影响

环境变量

LC_CTYPE 用于确定将文件内的文本解释为单字节和（或）多字节字符。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或 **LC_MESSAGES** 或将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 。

如果任一国际化变量中包含无效设置，则 **head** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

警告

输入行的长度被限制为 {**LINE_MAX**} 字节。

另请参阅

tail(1)、 cat(1)、 more(1)、 pg(1)。

符合的标准

head：SVID3、XPG4、POSIX.2

名称

host - DNS 查找实用程序

概要

host [**-aCdlnrTwv**] [**-c** *class*] [**-N** *ndots*] [**-R** *number*] [**-t** *type*] [**-W** *wait*] *name server*

说明

host 是一种用于执行 DNS 查找的简单实用程序。该命令通常用于将名称转换为 IP 地址或将 IP 地址转换为名称。当未指定参数或选项时， **host** 将输出其命令行参数及选项的简短摘要。

参数

name 指要被查找的域名。此参数也可以是由点分十进制的数所组成的 IPv4 地址或者由冒号分隔的 IPv6 地址，这时， **host** 在缺省情况下将执行该地址的反向查找。

server 这是一个可选参数，其值是 **host** 应该查询的名称服务器的名称或 IP 地址，而不是在 **/etc/resolv.conf** 中列出的一个或多个服务器。

选项

a 该选项等效于设置 **v** 选项并请求 **host** 进行 **ANY** 类型的查询。

C 使用该选项时， **host** 将尝试显示为 *name* 区域列出的所有授权名称服务器上针对该区域的 SOA 记录。名称服务器的列表是按照为该区域找到的 NS 记录来定义的。

c *class* 此选项指示 **host** 进行由 *class* 指定的类别的 DNS 查询。该选项可用于查询 Hesiod 或 Chaosnet 类别资源记录。缺省类别是 **IN** ，表示 Internet。

d **v** 当使用这两个选项之一时， **host** 将生成详细输出。这两个选项是等效的。它们可实现向后兼容。在以前的版本中， **d** 选项可切换调试跟踪，而 **v** 可启用详细输出。

l 该选项可选择列表模式。它使 **host** 执行 *name* 区域的区域传输。提供该参数是为了与以前的实现相兼容。该选项等效于进行 **AXFR** 类型的查询。

n 该选项指定在进行 IPv6 地址的反向查找时，应该使用在 RFC1886 中定义的 IP6.INT 域和“nibble”标记。缺省情况下使用 RFC2874 中定义的 IP6.ARPA 及二进制标记。

N *ndots* 该选项可设置要使 *name* 被视为绝对名称，该名称中必须存在的点的数目。可使用 **/etc/resolv.conf** 中的 **ndots** 语句来定义其缺省值，当不存在 **ndots** 语句时，缺省值为 1。点数少于该值的名称将被解释为相对名称，并会在 **/etc/resolv.conf** 的 *search* 或 *domain* 指令所列出的域中进行搜索。

R *number* 使用该选项可以更改 UDP 重试查找的次数。 *number* 表示 **host** 在没有获得答复时重复查询的次数。缺省的重试次数为 1。如果 *number* 为负数或零，则重试次数将缺省为 1。

r 通过该选项可以进行非递归查询。设置该选项会清除 **host** 查询中的 **RD** （需要递归）位。这意味着接收该查询的名称服务器不会试图解析 *name* 。

该选项可通过进行非递归查询，并期待收到对查询的答复来使 **host** 模仿名称服务器的操作，这些答复通常涉及到其他名称服务器。

T 该选项使 **host** 在查询名称服务器时使用 TCP 连接。对于需要 TCP 的查询，例如区域传输 (AXFR) 请求，会自动选择 TCP。在缺省情况下， **host** 进行查询时使用 UDP。

t *type* 可使用该选项选择查询类型。 *type* 可以是任何可识别的查询类型： **CNAME** 、 **NS** 、 **SOA** 、 **SIG** 、 **KEY** 、 **AXFR** 等等。未指定查询类型时， **host** 会自动选择一种适当的查询类型。缺省情况下，它会查找 A 记录，但是如果指定了 **C** 选项，则会查询 SOA 记录；而如果 **name** 是由以点分隔的十进制数组成的 IPv4 地址或以冒号分隔的 IPv6 地址，则 **host** 会查询 PTR 记录。

W *wait* 可通过 **W** 选项来控制等待响应的时间。该选项使 **host** 等待 *wait* 指定的秒数。如果 *wait* 小于 1，则等待的间隔被设置为 1 秒。

w 当使用了该选项时， **host** 将永远有效地等待响应。等待响应的时间将设置为由硬件的最大整数值指定的秒数。

文件

/etc/resolv.conf

另请参阅

named(1M)、 resolver(4)。

名称

hostname - 设置或显示当前主机系统的名称

概要

hostname [*name_of_host*]

说明

hostname 命令用于显示当前主机的名称，与 **gethostname()** 系统调用结果相同（请参阅 *gethostname*(2)）。拥有相应权限的用户可通过指定参数 *name_of_host* 来设置主机名；通常在启动脚本 **/sbin/init.d/hostname** 中完成此操作。 *name_of_host* 参数限制为 **MAXHOSTNAMELEN** 个字符，如 **<sys/param.h>** 中所定义。

如果支持网络产品，则可能使用其他名称识别系统。请参阅随您系统附带的节点管理器文档资料。

警告

如果指定了 *name_of_host* 参数，则生成的主机名更改仅在系统重新引导后才有效。要永久地更改主机名，请运行专用初始化脚本 **/sbin/set_parms** （请参阅《Using Your HP Workstation》）。

HP-UX 支持多种类型的网络服务，每种网络服务都单独使用一种分配的系统名和命名约定。要确保系统行为可预知，关键是用某种方式分配系统名（也称为主机名或节点名），这种方式应保证各种网络设备之间交互操作时，分配的系统名不会发生冲突。

系统不依赖于特定位置的单个系统名，部分原因是不同的服务使用如下所述的不同名称格式。 **hostname** 和 **uname** 命令以如下方式分配系统名：

节点名	命令	名称格式	使用方
Internet 名称	**hostname** 名称	*sys*[*.x.y.z...*]	ARPA 和 NFS 服务
UUCP 名称	**uname -S** 名称	*sys*	**uucp** 和相关程序

其中， *sys* 表示分配的系统名。 强烈建议所有命令和位置中都使用相同的 *sys* ，可选项 *.x.y.z...* 遵循特定 RPA/NFS 环境的指定表示法。

Internet 名称经常称为主机名或域名（与 NFS 域名不同）。有关 Internet 命名约定的详细信息，请参考 *hostname*(5) 。

只要在任何文件中或使用上述任何命令更改了系统名，都应在所有其他位置中进行同样的更改。除上述文件或命令之外的其他文件或命令（例如用于避开 **uname** 时使用的 **/etc/uucp/Permissions** ）可能包含或更改系统名。要确保操作正确，它们也应使用相同的系统名。

通常，系统名由 **/sbin/init.d/hostname** 脚本在启动时进行分配，且不应在其他位置更改。

作者

hostname 由加州大学伯克利分校开发。

另请参阅

uname(1)、 gethostname(2)、 sethostname(2)、 uname(2)、 hostname(5)。

《Using Your HP Workstation》

名称

hp - 处理 HP 2640 和 HP 2621 系列终端的特殊功能

概要

hp [**-e**] [**-m**]

说明

hp 支持 Hewlett-Packard HP 2640 和 HP 2621 系列终端的特殊功能，主要目的是为了生成大多数 **nroff** 输出的精确表示形式。典型用法如下：

nroff -h *files* ... **| hp**

无论给定终端上具有何种硬件选件， **hp** 尝试执行对下划线和反向换行符执行适当的操作。如果终端具有“显示增强”功能，则下标和上标将明确显示。如果它具有“数学符号”功能，则可以显示希腊字符和其他特殊字符。

选项

hp 可识别下列选项：

-e 指明终端具有“显示增强”功能，以充分利用所添加的显示模式。重叠字符以下划线模式显示。上标以半亮度模式显示，下标以半亮度模式和下划线模式显示。如果忽略此标志，则 **hp** 认为终端缺少“显示增强”功能。在这种情况下，所有重叠字符、下标和上标均以反色显示；即，亮中黑，而不是黑中亮。

-m 通过删除换行符来请求输出最小化。将 3 个或更多个换行符的任何连续序列转换为只有 2 个换行符的序列；即，任何数量的连续空白行均只生成单个空白输出行。这样可以在屏幕上保留更多实际文本。

诊断信息

line too long

超过 1,024 个字符的行的表示形式。

返回值

hp 在正常终止时返回零，对于所有错误均返回 2。

警告

将“叠印序列”定义为其后紧跟退格（此退格后紧跟另一输出字符）的输出字符。在这样的序列中，如果两个输出字符之一为下划线，则另一输出字符以加下划线或反色显示；否则，只显示第一个输出字符（再以下划线或反色显示）。如果退格出现在 ASCII 控制字符旁边，则不进行任何特殊操作。控制字符（例如，反向换行符和退格）序列可以使文本“消失”；尤其是，由 **tbl** 生成的、包含竖线的表经常丢失包含竖线“底部”的文本行，除非对 **hp** 的输入采取通过 **col** 进行管道输入（请参阅 *col*(1)）。

尽管一些终端确实支持数字上标字符，但并未尝试过显示这些字符。

另请参阅

col(1)、neqn(1)、nroff(1)、tbl(1)。

名称

hyphen - 查找用连字符连接的单词

概要

hyphen [*files*]

说明

hyphen 在 *files* 中查找所有位于行尾的用连字符连接的单词，并将其显示在标准输出中。如果没有指定参数，则会使用标准输入；因此， **hyphen** 可以用作过滤器。

举例

为 *textfile* 准备 **nroff** 连字符检查文件。

mm textfile | hyphen

警告

hyphen 不能处理用连字符连接的 斜体（即带下划线的）单词；通常会完全忽略或破坏它们。

hyphen 偶尔会使人感到迷惑，但除了输出多余的内容外不会造成任何不良影响。

另请参阅

mm(1)、 nroff(1)。

名称

iconv - 字符代码集转换

概要

iconv -f *fromcode* **-t** *tocode* [*file*]...

说明

iconv 将输入文件中字符的编码从 *fromcode* 代码集转换为 *tocode* 代码集，并在标准输出上写入结果。如果未提供输入文件， **iconv** 将从标准输入中读取。如果 **-** 用作输入文件名， **iconv** 将在该点处读取标准输入。 **--** 可用于界定选项的结束（请参阅 *getopt*(3C) ）。

选项

iconv 可识别下列选项:

-f *fromcode* 将与选项参数 *fromcode* 对应的代码集标识为输入将从其进行转换的代码集。

-t *tocode* 将与选项参数 *tocode* 对应的代码集标识为输入将转换到的代码集。

fromcode 和 *tocode* 名称可以是 iconv 配置文件 **/usr/lib/nls/iconv/config.iconv** 中列出的任意基名和别名。有关支持的代码集名称列表的详细信息和配置文件，请参阅 *iconv*(3C) 。

外部语言环境影响

环境变量

LANG 为未经设置或设置为空的国际化变量提供了缺省值。如果 **LANG** 未设置或设置为空，则会使用缺省值“C”（请参阅 *lang*(5) ）。如果任一国际化变量中包含无效设置，则 **iconv** 的行为类似于所有国际化变量都设为“C”。请参阅 *environ*(5) 。

LC_ALL 如果设为非空字符串值，则会覆盖所有其他国际化变量的值。

LC_CTYPE 确定文本作为单字节和（或）多字节字符的解释、字符作为可打印字符的分类以及在常规表达式中按字符类表达式匹配的字符。在转换文件的过程中，该变量通过使用 *fromcode* 选项参数来替代。

LC_MESSAGES 确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息以及写入到标准输出的信息消息的格式和内容。

NLSPATH 确定消息目录的位置，以便处理 **LC_MESSAGES** 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

警告

如果输入字符在通过 **-t** 选项选择的代码集（“目标”代码集）中没有有效的等效项，则将映射到“活版字符”（如果已经为该转换定义）。（请参阅 *genxlt*(1) 和 *iconv*(3C) ）。

如果输入字符不属于通过 **-f** 选项选择的代码集（“源”代码集），命令将终止。

举例

将文件 **foo** 的内容从代码集 Roman8 转换为 ISO 8859/1，然后将结果存储在文件 **bar** 中。

iconv -f roman8 -t iso8859_1 foo > bar

文件

/usr/lib/nls/iconv/config.iconv iconv 配置文件

作者

iconv 由 HP 开发。

另请参阅

getopt(3C)、 iconv(3C)。

符合的标准

iconv：XPG2、XPG3、XPG4

名称

id - 输出用户和组的 ID 及名称

概要

id [**-u**] [**-nr**] [*user*]

id [**-g**] [**-nr**] [*user*]

id [**-G**] [**-n**] [*user*]

id [**-P**]

说明

id 命令可将消息写入标准输出，针对进程给出用户和组的 ID 及名称。如果有效 ID 与实际 ID 不同，那么两者都会被输出。

如果进程具有关联的补充组（请参阅 *groups*(1)），则关联的补充组也被写入。

如果指定了 *user* 操作数，则选定用户的用户 ID 和组 ID 也被写入。在这种情况下，假定有效 ID 与实际 ID 相同。

选项

使用下列选项可以修改上述行为。

-g 仅显示组 ID。缺省值是有效组 ID；要进行修改，请使用 **-r** 选项。如果进程具有补充组关联，而其不同于有效组 ID 或实际 ID（如果使用了 **-r** 选项），则将在同一行上显示所有此类关联。缺省值为十进制格式；要进行修改，请使用 **-n** 选项。

-G 使用“%u\n”格式，仅输出所有不同的组 ID（有效 ID、实际 ID 和补充 ID）。如果存在多个不同的组关联，则在输出 <换行> 之前，使用“ %u”格式输出所有此类关联。

-n 使用 **-u** 、 **-g** 或 **-G** ，显示 ID 名称而不是 ID 数。

-r 使用 **-u** 、 **-g** 或 **-G** ，显示实际 ID 而不是有效 ID。

-u 仅显示用户 ID。缺省值是有效用户 ID；要进行修改，请使用 **-r** 选项。缺省值为十进制格式；要进行修改，请使用 **-n** 选项。

-P 显示进程的进程资源组 ID，以及用户和组的 ID 及名称。 **-P** 选项会忽略用户参数。请参阅“相关内容”中的“HP Process Resource Manager”。

返回值

由 **id** 返回的错误代码包括：

0 成功。
1 未找到用户，或选项无效，或选项组合无效。
2 在不支持 PRM 或 PRM 未配置的情况下，给定了 **-P** 选项。

举例

要显示当前用户或组的数据:

id

生成:

uid=1834(allanp) gid=20(users)

要显示当前进程的组 ID 数:

id -g

生成:

20

要显示当前进程的组名:

id -gn

生成:

users

要显示另一用户的用户和组数据:

id ralford

生成:

uid=329(ralford) gid=20(users)

要显示当前进程的 PRM 组 ID:

id -P

生成:

uid=329(ralford) gid=20(users) prmid=1(OTHERS)

相关内容

HP Process Resource Manager

要使用 **-P** 选项，则必须安装可选的 HP Process Resource Manager (PRM) 软件，并进行配置。有关如何配置 HP PRM 的说明，请参阅 *prmconfig*(1) ；有关进程资源组定义的详细信息，请参阅 *prmconf*(4) 。

作者

id 由 HP 和 AT&T 开发。

另请参阅

groups(1)、 logname(1)、 getuid(2)。

HP Process Resource Manager： 《HP Process Resource Manager User's Guide》中的 prmconfig (1)、 prmconf (4)。

符合的标准

id： SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

ident - 识别 RCS 中的文件

概要

ident *file* ...

说明

ident 在已命名的文件中搜索模式 **$***keyword***:...$** 的所有发生实例，其中 *keyword* 可以为：

Author	**Log**
Date	**Revision**
Header	**Source**
Locker	**State**

这些模式通常由 RCS **co** 命令自动插入，但也可以手动插入（请参阅 *co*(1)）。

ident 可处理文本文件和对象文件。例如，如果文件 **f.c** 中的 C 程序包含：

```
char rcsid[] = "$Header:  Header information $";
```

并且 **f.c** 被编译为 **f.o**，则命令：

```
ident f.c f.o
```

输出：

```
f.c:
    $Header: Header information $

f.o:
    $Header: Header information $
```

作者

ident 由 Walter F. Tichy 开发。

另请参阅

ci(1)、co(1)、rcs(1)、rcsdiff(1)、rcsintro(5)、rcsmerge(1)、 rlog(1)、rcsfile(4)。

名称

idlookup - 识别特定 TCP 连接的用户

概要

idlookup *host-or-ip-number local-port foreign-port*

说明

idlookup 可用于识别 TCP 连接的远程终端的用户，假定另一端的主机正在运行标识服务器。

host-or-ip-number 是位于连接另一端的主机的名称或其 IP 地址。

local-port 和 *foreign-port* 都是位于连接两端的端口号，或端口的服务名。

警告

请注意，对 *local-port* 和 *foreign-port* 的引用遵循 RFC931 中的术语，并且是从 服务器而非用户的角度引用的。

作者

idlookup 最初由 Peter Eriksson 编写。本手册页最初由 Dave Sheild 编写。

另请参阅

sendmail(1M)、 identd(1M)、 RFC931。

名称

ied - 交互式程序的输入编辑器和命令历史记录

概要

ied [**-dirt**] [**-h** *file*] [**-s** *size*] [**-p** *prompt*] [**-k** *charmap*] *utility* [*arguments* ...]

说明

ied 是一个实用程序命令，它旨在充当用户和交互式程序（例如 bc、bs 或 Shell）之间的一个界面，以提供 Korn Shell 中的大部分行编辑和历史记录功能。 **ied** 将 *utility* 的名称解释为要执行的命令，并将 *arguments* 作为参数传递给实用程序。 *utility* 的后续输入随后可以访问编辑功能和历史记录功能，这与 *ksh* 提供的功能很相似。

ied 监视它用来运行命令的 pty 的状态， **ied** 启动后，当所运行应用程序的状态由 tty 状态更改为其他状态时， **ied** 就变为“透明的”。它允许程序将 Shell 转义为屏幕智能程序。一般来说， **ied** 不应该以任何方式对它提供前端的任何程序的操作进行干预。这包括 Korn Shell 本身：在这种情况下， **ied** 会提供由 **ksh** 运行的任何应用程序的历史记录，并且 **ksh** 会提供它自己的独立历史记录。在极为有用的情况下， **ied** 可以用作登录 Shell（可能是 **ksh** 或 **csh** ）的前端。在这种情况下，所有使用一般行编辑功能的应用程序会获取行编辑和历史记录，从而共享单个历史记录。如果 Shell 提供这样的机制，则它会继续拥有自己的独立历史记录。

当 **ied** 处于其透明模式时，不保存任何历史记录。特别是， **vi** 的 **ex** 模式不使用一般的行编辑功能（确切地说，是模拟行编辑），在这种情况下， **ied** 无法提供历史记录。也无法使用 **ied** 编辑 **Subject:** 和 mailx 的地址行编辑。

选项

多个选项和命令行参数控制 **ied** 的操作：

-d 调试模式。输出有关程序操作的信息。最好用它来确定程序是否意外地将 **ied** 置于透明模式下。

-h *filename* 将历史记录保存在一个名为 *filename* 的文件中。如果已存在同名文件，并且该文件是一个历史记录文件，则它的后面部分（ **-s** 选项指定的最后 *size* 的行）将用作历史记录的初始值。如果没有使用 **-h** 选项，则环境变量 **IEDHISTFILE** 用于提供文件名。如果两者均未提供，则使用一个未命名的临时文件，并且不提供初始值。

-i 强制交互模式。通常，当标准输入不是 tty 时， **ied** 只是通过 **exec** 执行请求其作为前端的命令（这允许 Shell 中使用的命令利用别名，而不会影响它们的操作）。该选项强制将 **ied** 保留为一个前端，同时所有的编辑功能都是正常的。它使得在交互模式下和批处理模式下表现不同的实用程序，可在交互模式下通过管道或文件驱动。这在测试产生该差别的命令时尤其有用。

-k *charmap* *charmap* 是一个等于或少于 256 行的文件。文件中的行号是 **ied** 输入所显示的字符的序数，而行上的字符是作为输出生成的字符（同时还用作编辑字符）。它允许对（普通）按键进行重新映射（例如 Dvorak 键盘的按键）。字符必须在每一行对应的一列中开始，并且表示为 1 到 4 个字符后跟一个空格，或是后接下一行的换行符形式。空格后的字符将作为注释被忽略。单字符条目表示其本身。第一个字符是插字符号 (^) 的双字符

条目，会将第二个字符转换为对应的控制字符。第一个字符是反斜杠 (\) 的双字符序列使用 C 语言约定：

\n	换行符	***\s***	空格
******	转义符	***\0***	空字符
\r	回车符	***\f***	换页符
\t	制表符	***\v***	纵向制表符
\b	退格		

三字符和四字符序列必须是 *nn* 或 *nnn*的形式，以便赋予字符八进制值。如果 *charmap* 的长度少于 256 行，则将剩余的字符映射为它们本身。

-p *prompt* 许多命令在输入就绪时并不进行提示。对于此类命令而言， **ied** 接近于一种提示机制。该机制并不总是完全成功的，但对很多命令是很有用的。在最坏的情况下，提示中散布有位于错误位置的输出。 *prompt* 是一个字符串，与 *printf*(3S) 的格式参数中所用的一样。可以包含的唯一 **%** 转换是： **%d** 的最多一个实例可转换为命令的序号， **%%** 的任意实例数可视为一个 **%** 字符。当 **ied** 在透明模式下进行操作时，将禁止显示提示。

-r 它设置"非原始"模式。通常，在读取简单文本时， **ied** 使用它自己的编辑功能。这使得 **ied** 在多数时候都使用 tty 行规则。缺省模式的缺点是，需要更多的上下文切换和常规处理。优点是 **ied** 更加透明。例如，要在非原始模式下特别发送一个文件结束标志，需要在文件结束标志字符（通常是 Ctrl-D）后面跟回车符。同样地，"下一个文字"功能 (Ctrl-V) 在非原始模式下无法对行清除和行终止功能进行转义。

-s *size* 该选项指定历史记录缓冲区的大小。如果 **ied** 启动时有一个现有的历史记录文件，则大约最后 *size* 行可用于历史记录机制（不保证该数字与 *size* 的值完全相等）。将保留文件中的其他行，直到对该历史记录文件再次启动了 **ied** ，并且文件大小大约超过 4 KB，此时 **ied** 将忽略文件开头的较旧条目，直到大小接近于 4 KB 为止。由于这仅在启动时发生，因此，历史记录文件在两次重新启动之间可以增长到非常大。更大的 *size* 值会使进程映像变得更大。

如果未指定 **-s** ，则使用环境变量 **IEDHISTSIZE** 的值。如果两者均未指定，则使用缺省值。

-t 设置透明模式。它强制 **ied** 永久地处于透明模式下（如上所述）。对于一些自动进行处理的类，该选项主要适于与 **-i** 一起使用。特别是，如果将某个命令作为输入（ **ied** 将其解释为编辑字符），则该选项对驱动此命令很有用。因此，使用 **-i** 和 **-t** 的适当组合，可以驱动一个编辑器（例如 **vi** ），或从批处理文件驱动一个屏幕智能应用程序。

ied 出错后， **SIGQUIT** 信号重复 3 次，通常会中止 **ied** 。但对于完全透明的应用程序，则必须从另一个窗口或终端终止 **ied** 。仅当无法指示使用中的进程自行终止时，它们才会真正关联。

ied 的编辑功能实质上是 **ksh** 中的功能。下面仅介绍了与 **ksh** 不同的功能。与 **ksh** 中一样，编辑的样式是通过环境变量 **VISUAL** 确定的，如果未指定 **VISUAL** ，则通过 **EDITOR** 确定。检查的值应当以 **vi** 、 **emacs** 或 **gmacs**

结尾，以指定编辑器的类型。如果不是这样的话， **ied** 不会进行编辑，并且历史记录无法访问。

在 vi 模式下：

J 连接行。将最近编辑的行（在将一行发送到应用程序后，它立即为空）视为历史记录的“最后一行”，历史记录中显示的当前行将被追加到最后一行的结尾，同时，在历史记录中所处的位置将重置为位于最后一行，随后会显示该行。在最后一行的新旧文本之间将插入一个空格。光标将停留在该空格处。由于 **ied** 对行延续性的了解极少，这对编辑长语句是很有用的。

v 不支持。

V 不支持。

不向应用程序发送任何内容，但在历史记录中插入行（在向历史记录文件中添加注释时很有用）。

<*esc*>**,*,=** （文件名扩展）。不支持。

@ 宏扩展。不支持。

但请注意， **ksh** 有一个很少使用的功能 _ ，它可以替换前一行中的文字（它不是宏 **$_** ，而是一个编辑器命令）。如果给定了前导 *count* ，它将使用最后一行中的第 *count* 个字。这对于 **ied** 更为有用。

在 emacs/gmacs 模式下：

M-*, **M-=**, **M-**<*esc*> （文件名扩展）不支持。

请注意，命令 **M-.** （及其同义项 **M-_** ）提供的功能与 vi 模式下的 _ 命令提供的功能相同。

Macro expansion. 不支持。

^O 虽然支持，但是在屏幕上可能始终无法正确显示。使用 **^L** 命令可以重绘行。有关提示的讨论，请参阅下文。

举例

在 **bc** 命令中添加交互式编辑：

ied bc

使用从 **script** 中提取的命令对 **testfile** 执行 **vi** ：

cat script | ied -i -t vi testfile

请注意，如果没有使用 **ied** ，则 **vi** 将会出现错误，这是因为它的标准输入不是一个终端设备。在这种情况下，不需要使用 **-t** ，因为 **vi** 将其自身置于原始模式下，但对于全然不同的应用程序而言，则可能需要使用 **-t** 。

命令行

ied -i -t grep '^x:' data_file | tee x_lines

搜索文件 **data_file** 来查找以 **x:** 开头的行，从而将一个副本发送到终端，将第二个副本发送到文件 **x_lines** ，如同以下命令行一样

grep '^x:' data_file | tee x_lines

两个命令行的区别在于，在未使用 **ied** 的命令行中， **grep** 直接写入管道，因此缓存了其输出。如果 **data_file** 很大并且与模式匹配的行不多，则终端上的输出将延迟。通过使用 **ied** ， **grep** 的输出更改为 pty 状态，这使得 **grep** 可以在准备就绪时输出每一行。

警告

由于 **ied** 无法了解每个应用程序的所有情况，因此，它可能会因与应用程序不协调的计时或提示而造成混淆。既然使用 **ied** 决不是必需的，那么应由用户确定是使用还是不使用 **ied** 对应用程序更为有利。但一般而言，不与 **ied** 相混淆的程序，通常也最有可能从它受益。

ied 在本身不提供活动提示符时，尝试凭直觉获取当前的活动提示符。不过，这并不总是能成功。即使成功， **ied** 的计时及使用中的命令有时也可能在输出中造成混淆。 **^L** 命令可以在 emacs 和 vi 模式下，采用与用于创建下一个命令相一致的方式重绘编辑行。

作者

ied 由 HP 开发。

另请参阅

ksh(1)。

名称

insertmsg - 使用 findstr 输出插入对 catgets() 的调用

概要

insertmsg [**-h**] [**-i***amount*] [**-n***number*] [**-s***number*] *stringlist*

说明

insertmsg 检查文件 *stringlist* ，该文件假定是经过后续编辑后已删除不需要本地化的所有字符串之后的 **findstr** 的输出（请参阅 *findstr*(1) ）。如果指定了 **-h** 选项， **insertmsg** 会将以下行放在 *stringlist* 中指定的每个文件的开头：

```
#ifndef NLS
#define catgets(i,sn,mn,s) (s)
#else NLS
#define NL_SETN number
#include <nl_types.h>
#endif NLS
```

其中 *number* 是由 **-s** 选项定义的集合编号；缺省值是 **1** 。对于 *stringlist* 中的每个字符串， **insertmsg** 都将用以下形式的表达式在对应文件中为字符串添加引号：

(catgets(catd,NL_SETN,*msg_num***,"***defaultstring***"))**

defaultstring 是由 *stringlist* 中的行引用的初始字符串， *msg_num* 将替换为分配给该字符串的消息号。分配的消息号以 **-n** 选项定义定义的数字开始，按照 **-i** 选项定义的数量递增。起始消息号和递增量的缺省值均是 **1** 。如果 *name***.c** 是在 *stringlist* 文件中指定的要修改的文件， *insertmsg* 会将修改后的源放入 **nl_***name***.c** 。这样，用户必须手动编辑文件 **nl_***name***.c** 插入以下语句：

```
nl_catd catd;
catd = catopen("appropriate message catalog",0);
```

数据类型 **nl_catd** 在 **<nl_types.h>** 中定义， **catd** 是对 **catgets()** 的调用的参数，必须为 *stringlist* 中的每个字符串插入这些调用。

insertmsg 还向标准输出发送一个可用作 **gencat** 输入的文件（请参阅 *gencat*(1) ）。

外部语言环境影响

环境变量

LC_CTYPE 用于确定将文本解释为单字节和（或）多字节字符的方法。

LC_MESSAGES 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或 **LC_MESSAGES** ，或者将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果 **LANG** 未指定或设置为空字符串，则将使用缺省的“C”而不是 **LANG** （请参阅 *lang*(5) ）。如果任一国际化变量中包含无效设置， **insertmsg** 会假定所有国际化变量都缺省为“C”。请参

阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

insertmsg exiting : lost in strings file

insertmsg 在字符串文件中需要左双引号或右双引号的地方找不到左双引号或右双引号。它将异常中止。请检查字符串文件，确保保留在那里的行未被更改。

警告

如果未使用 **-h** 选项，则可能必须将以下语句手动添加到由 **insertmsg** 创建的文件：

#include <nl_types.h>

insertmsg 将一个指针插入在每个调用上覆盖的静态区域。

insertmsg 命令是 HP 专有程序，不可移植到其他供应商的系统上，并且将不在将来的 HP-UX 版本中提供。

作者

insertmsg 由 HP 开发。

另请参阅

findstr(1)、 gencat(1)、 catgets(3C)、 catopen(3C)。

名称

iostat - 报告 I/O 的统计信息

概要

iostat [**-t**] [*interval* [*count*]]

说明

iostat 迭代地报告系统中各活动磁盘的 I/O 统计信息。磁盘数据以 4 列格式进行组织排列:

列标题	解释
device	设备名称
bps	每秒传输的字节数 (KB)
sps	每秒执行的寻道数
msps	平均每次寻道所用的毫秒数

如果存在两个或更多个磁盘，则各磁盘数据通过连续的行来显示。

要计算这一信息，需要计算各磁盘的寻道、数据传输完成次数以及传输的单词数等。另外，每秒将检查各磁盘的状态 **HZ** 次（如 **<sys/param.h>** 中定义），若磁盘处于活动状态，则生成一个标签。这些数字可与各设备的传输速率结合使用，以确定各设备的平均寻道时间。

随着磁盘新技术的出现（例如数据条带化，可将一次数据传输分布在多个磁盘上进行），已不可能精确计算平均每次寻道所用的毫秒数。最好的情况下，能得到的也仅仅是近似值而已，并且该值根据一些具体动态系统环境的不同还会有很大的变化。出于这个原因，同时也为了维护向后兼容性，平均每次寻道所用的毫秒数 (**msps**) 字段值被设置为 1.0。

选项

iostat 可识别下列选项和命令行参数:

-t 报告终端统计信息以及磁盘统计信息。终端统计信息包括:

- **tin** 从终端读取的字符数。
- **tout** 写入终端的字符数。
- **us** 系统（活动处理器）处于用户模式下的时间百分比。
- **ni** 系统（活动处理器）在用户模式下运行低优先级 (*nice*) 进程所花费的时间百分比。
- **sy** 系统（活动处理器）处于系统模式下的时间百分比。
- **id** 系统（活动处理器）处于空闲状态的时间百分比。

interval 显示作为上一个 *interval* 秒的总计的连续行。所报告的第一行用于显示自重新引导后过去的时间，每个后续行则仅用于上一次间隔。

count 重复统计 *count* 次。

举例

显示所有磁盘当前 I/O 统计信息：

iostat

每 10 秒显示所有磁盘的 I/O 统计信息，直至按下 INTERRUPT 或 QUIT 字符为止：

iostat 10

每 10 秒显示所有磁盘的 I/O 统计信息，并在连续读取 5 次后终止：

iostat 10 5

每 10 秒显示所有磁盘的 I/O 统计信息，同时展示终端和处理器的统计信息，并在连续读取 5 次后终止：

iostat -t 10 5

警告

iostat 的用户不能依赖于其输出中具体的字段宽度以及分隔形式，因为它们可能会根据系统（HP-UX 的版本）及要显示的数据的不同而异。

作者

iostat 由加州大学伯克利分校和 HP 联合开发。

文件

/usr/include/sys/param.h

另请参阅

vmstat(1)。

名称

ipcrm - 删除消息队列、信号量集或共享内存的标识符

概要

ipcrm [*option*]...

说明

ipcrm 命令可删除一个或多个指定的消息队列、信号量集或共享内存的标识符。

选项

标识符由以下 *option* 指定：

-m *shmid* 从系统中删除共享内存标识符 *shmid* 。最后一次分离后，与其相关联的共享内存段及数据结构都将遭到破坏。

-q *msqid* 从系统中删除消息队列标识符 *msqid* ，并破坏与其相关的消息队列和数据结构。

-s *semid* 从系统中删除信号量标识符 *semid* ，并破坏与其相关联的信息量集和数据结构。

-M *shmkey* 从系统中删除由键 *shmkey* 创建的共享内存标识符。最后一次分离后，与其相关联的共享内存段及数据结构都将遭到破坏。

-Q *msgkey* 从系统中删除由键 *msgkey* 创建的消息队列标识符，并破坏与其相关联的消息队列和数据结构。

-S *semkey* 从系统中删除由键 *semkey* 创建的信号量标识符，并破坏与其相关联的信号量集和数据结构。

msgctl(2) 、 *shmctl*(2) 和 *semctl*(2) 中详细描述了删除。使用 **ipcs** （请参阅 *ipcs*(1) ）可以找到标识符和键。

另请参阅

ipcs(1) 、 msgctl(2) 、 msgget(2) 、 msgop(2) 、 semctl(2) 、 semget(2) 、 semop(2) 、 shmctl(2) 、 shmget(2) 、 shmop(2) 。

符合的标准

ipcrm：SVID2、SVID3

名称

ipcs - 报告进程间通信设备的状态

概要

ipcs [**-mqs**] [**-abcopt**] [**-C** *core*] [**-N** *namelist*]

说明

ipcs 用于显示有关活动的进程间通信设备的特定信息。如果不使用选项，则 ipcs 会以短格式显示有关系统中当前活动的消息队列、共享内存段和信号量的信息。

选项

下列选项会限制显示相应的设备。

(none)	它等效于 **-mqs** 。
-m	显示有关活动共享内存段的信息。
-q	显示有关活动消息队列的信息。
-s	显示有关活动信号量的信息。

下列选项可在显示中添加数据列。请参阅下面的“列说明”。

(none)	显示缺省列：针对所有设备： **T** 、 **ID** 、 **KEY** 、 **MODE** 、 **OWNER** 、 **GROUP** 。
-a	显示所有适当的列。此选项等效于 **-bcopt** 。
-b	显示所允许的最大量的信息：针对消息队列： **QBYTES** ；针对共享内存段： **SEGSZ**；针对信号量： **NSEMS** 。
-c	显示创建者的登录名和组名：针对所有设备： **CREATOR** 、 **CGROUP** 。
-o	显示有关未完成的使用的信息：针对消息队列： **CBYTES** 、 **QNUM** ；针对共享内存段： **NATTCH** 。
-p	显示进程数量的信息：针对消息队列： **LSPID** 、 **LRPID** ；针对共享内存段： **CPID** 、 **LPID** 。
-t	显示时间信息：针对所有设备： **CTIME** ；针对消息队列： **STIME** 、 **RTIME** ；针对共享内存段： **ATIME** 、 **DTIME** ；针对信号量： **OTIME** 。

下列选项重新定义了信息源。

-C *core*	使用 *core* 替换 **/dev/kmem** 。 *core* 可以是一个核心文件，也可以是 **savecrash** 或 **savecore** 创建的一个目录。
-N *namelist*	使用文件 *namelist* 或 *core* 中的 *namelist* 来替换 **/stand/vmunix** 。它会打开一个崩溃转储进行读取。有关详细信息，请参考 cr_open*(3)* 。

列说明

下面给出了 **ipcs** 列表中的列标题以及各列的含义。这些列以下面所示的顺序从左到右打印。

T 设备类型:

- **m** 共享内存段
- **q** 消息队列
- **s** 信号量

ID 设备条目的标识符。

KEY 用作 **msgget()** 、 **semget()** 或 **shmget()** 的参数以创建设备条目的键。(注释:当共享内存段删除后,该内存段的键会更改为 **IPC_PRIVATE** ,直到该内存段上连接的所有进程都与其分离)。

MODE 设备访问模式和标志:该模式由 11 个字符组成,这些字符解释如下:

前两个字符可以是:

- **R** 在 **msgrcv()** 上有一个进程在等待。
- **S** 在 **msgsnd()** 上有一个进程在等待。
- **D** 关联的共享内存段已删除。当该内存段连接的最后一个进程与其分离后,该标志将消失。
- **C** 当执行第一个连接时,关联的共享内存段将被清除。
- **-** 相应的专用标志未设置。

后面 9 个字符按每组三个、分成三组进行解释。第一组表示所有者的权限,第二组表示设备条目组中的其他人的权限,最后一组表示其他所有人的权限。

每组中,第一个字符表示读取权限,第二个字符表示写入或更改设备条目的权限,最后一个字符当前未使用。

- **r** 授予读取权限。
- **w** 授予写入权限。
- **a** 授予更改权限。
- **-** 未授予所指示的权限。

OWNER 设备条目所有者的登录名。

GROUP 设备条目所有者组的组名。

CREATOR 设备条目创建者的登录名。

CGROUP 设备条目创建者组的组名。

CBYTES 关联的消息队列中当前未完成的消息的字节数。

QNUM 关联的消息队列中当前未完成的消息数。

QBYTES 关联的消息队列中未完成的消息所允许的最大字节数。

LSPID 向关联的消息队列发送消息的最后一个进程的进程 ID。

LRPID 从关联消息队列接收消息的最后一个进程的进程 ID。

STIME 将最后一条 **msgsnd()** 消息发送到关联消息队列的时间。

RTIME 从关联的消息队列接收到最后一条 **msgrcv()** 消息时的时间。

CTIME 创建或更改关联设备条目的时间。

NATTCH 关联共享内存段所连接的进程数。

SEGSZ 关联的共享内存段的大小。

CPID 共享内存段的创建进程的进程 ID。

LPID 与共享内存段连接或分离的最后一个进程的进程 ID。

ATIME 关联共享内存段的最后一个 **shmat()** 连接的完成时间。

DTIME 关联共享内存段的最后一个 **shmdt()** 分离的完成时间。

NSEMS 与信号量条目关联的组中的信号量的数目。

OTIME 在与信号量条目关联的组上，最后一个 **semop()** 信号量操作的完成时间。

警告

ipcs 仅生成实际系统状态的大概表示，因为在 **ipcs** 获取所请求的信息时，系统进程仍在不断变化。

请不要局限于输出的确切字段宽度和间距，因为它们会根据系统、HP-UX 的发行版及要显示的数据的变化而变化。

文件

/dev/kmem 内核虚拟内存
/etc/group 组名
/etc/passwd 用户名
/stand/vmunix 系统名称列表

另请参阅

msgop(2)、 semop(2)、 shmop(2)。

符合的标准

ipcs：SVID2、SVID3

名称

join - 关系数据库运算符

概要

join [*options*] *file1 file2*

说明

join 在标准输出中生成由 *file1* 和 *file2* 的行指定的两个关系的联接。如果 *file1* 或 *file2* 是 **-** ，则使用标准输入。

file1 和 *file2* 必须按照联接时所依据的字段，以递增排序序列（请参阅下面的“环境变量”）进行排序；该字段通常为每行中的第一个字段。

对于具有相同联接字段的 *file1* 和 *file2* 中的每对行，输出中只包含一行。输出行通常包含：公共字段，后跟来自 *file1* 的行的其余部分，然后是来自 *file2* 的行的其余部分。

缺省的输入字段分隔符是空格、制表符或换行符。在这种情况下，多个分隔符算作一个字段分隔符，且忽略前导分隔符。缺省的输出字段分隔符是空格。

下列选项中的某些选项使用参数 *n* 。此参数应该是 **1** 或 **2** （分别指 *file1* 或 *file2* ）。

选项

-a *n* 除了正常输出外，还将为文件 *n* 中每个不成对的行生成一行，其中 *n* 是 **1** 或 **2** 。

-e *s* 用字符串 *s* 替换空的输出字段。

-j *m* 基于两个文件的字段 *m* 进行联接。参数 *m* 必须用空格字符分隔。提供此选项和下面的两个选项是为了向后兼容。要获得可移植性，建议使用 **-1** 和 **-2** 选项（请参阅下文）。

-j1 *m* 基于 *file1* 的字段 *m* 进行联接。

-j2 *m* 基于 *file2* 的字段 *m* 进行联接。

-o *list* 每个输出行都包含 *list* 中指定的字段，列表中每个元素的形式是 *n***.***m* ，其中 *n* 是文件编号， *m* 是字段编号。除非特别要求，否则不输出公共字段。

-t *c* 使用字符 *c* 作为分隔符（制表符）。一行中出现的每个 *c* 都是有意义的。字符 *c* 用作输入以及输出的字段分隔符。

-v *file_number*
仅为 *file_number* 指定的文件中的每个不成对行生成一行，而不是生成缺省输出，其中 *file_number* 是 **1** 或 **2** 。

-1 *f* 基于文件 1 的字段 *f* 进行联接。字段从 1 开始编号。

-2 *f* 基于文件 2 的字段 *f* 进行联接。字段从 1 开始编号。

外部语言环境影响

环境变量

LC_COLLATE 可确定 **join** 期望的输入文件的排序序列。

LC_CTYPE 可确定将替代的空白字符作为输入字段分隔符，将文件内的数据的解释为单字节字符和（或）多字节字符。 **LC_CTYPE** 还可确定通过 **-t** 选项定义的分隔符是单字节字符还是多字节字符。

如果未在环境中指定 **LC_COLLATE** 或 **LC_CTYPE** ，或者将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5)）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **join** 就会认为所有国际化变量都设置为“C”（请参阅 *environ*(5) ）。

国际代码集支持

支持单字节字符代码集和多字节字符代码集，例外情况是不支持多字节字符文件名。

举例

以下命令行将联接口令文件和组文件，按数字组 ID 进行匹配，并输出登录名、组名和登录目录。假定已经按 **LC_COLLATE** 或 **LANG** 环境变量定义的排序序列，基于组 ID 字段对这些文件进行了排序。

join -1 4 -2 3 -o 1.1 2.1 1.6 -t: /etc/passwd /etc/group

以下命令将生成的输出中将包含：在两个已排序文件 *sf1* 和 *sf2* 中具有相同第一个字段的行的所有可能组合，而每行都包含来自 **sorted_file1** 的第一个和第三个字段以及来自 **sorted_file2** 的第二个和第四个字段：

join -j1 1 -j2 1 -o 1.1,2.2,1.3,2.4 sorted_file1 sorted_file2

警告

对于缺省的字段分隔，排序序列与 **sort -b** 的相同；对于 **-t** ，排序序列与普通排序的相同。

join 、 **sort** 、 **comm** 、 **uniq** 和 **awk** 的约定是不一致的。

当紧接在列出的文件名之前使用 **-o** 选项时，数字文件名可能会导致冲突。

作者

join 由 OSF 和 HP 开发。

另请参阅

awk(1)、 comm(1)、 sort(1)、 uniq(1)。

符合的标准

join ：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

kdestroy - 销毁 Kerberos 凭证

概要

kdestroy [**-q**] [**-c** *cache_filename*]

说明

kdestroy 实用程序可销毁用户的活动 Kerberos 授权凭证，方法是向包含这些凭证的指定凭证缓存中写入零。如果未指定凭证缓存，则将销毁缺省的凭证缓存。

选项

-q 静默运行。通常，当 **kdestroy** 不能成功销毁用户的凭证时，会发出嘟嘟的响声。 **-q** 标志会禁止此行为。

-c *cache_filename* 使用 *cache_filename* 作为凭证缓存的名称和位置。如果未使用该选项，则将使用缺省缓存的名称和位置。

缺省凭证缓存随系统的不同而变化。如果设置了 **KRB5CCNAME** 环境变量，则此变量的值将用于命名缺省凭证缓存。

大多数安装程序会建议您将 **kdestroy** 命令放在 **.logout** 文件中，这样，当您注销时会自动销毁您的凭证。

注释

对于 DCE 操作，请使用 **/opt/dce/bin/kdestroy** 。

环境

kdestroy 使用以下环境变量：

KRB5CCNAME 凭证缓存的位置。

警告

只有指定凭据缓存中的凭证会被销毁。使用了单独的凭证缓存来分别存放根实例和口令更改凭证。即使这些缓存应该销毁。建议将所有的用户凭证保留在单个凭证缓存中。

文件

/tmp/krb5cc_{*uid*} 缺省凭证缓存。 {*uid*} 是用户的十进制 UID。

作者

kdestroy 由麻省理工学院开发。

另请参阅

kinit(1)、 klist(1)、 kerberos(5)。

名称

kermit - 用于串行和网络连接的 C-Kermit 8.0 通信软件连接：调制解调器拨号、文件的传输和管理、终端连接、字符集转换、数字和字母的分页及脚本编程。

概要

kermit [*command-file*] [*options ...*]

说明

Kermit 是哥伦比亚大学在 Kermit 项目中开发的一系列用于文件传输、管理和通信的软件程序，适用于大多数的计算机和操作系统。用于 HP-UX 的 Kermit 版本被称为 “**C-Kermit**”，它既支持串行连接（直接连接或拨号连接），也支持 TCP/IP 连接。

可以将 C-Kermit 视为 **cu**、 **tip**、 **uucp**、 **ftp**、 **telnet**、 **rlogin**、 **expect** 的一个用户友好且功能强大的替代方案，甚至可以替代 Shell；也可以将其视为一个用于网络通信和串行通信的程序包，提供其他程序包所不具备的自动化、便捷、以及语言特性，且与其同类产品具有大量的共同属性，如用于其他 UNIX 平台的 C-Kermit；用于 Windows 95、Windows 98、Windows NT 和 2000，以及 OS/2 的 Kermit 95；用于使用 DOS 和 Windows 3.x 的 PC 的 MS-DOS Kermit；以及用于 VM/CMS、MVS/TSO 和 CICS 的 IBM Mainframe Kermit-370。C-Kermit 本身也可运行在 Digital VMS、Data General AOS/VS、 Stratus VOS、OS-9、QNX、Plan 9、Commodore Amiga，以及其他系统上。总之，C-Kermit、 Kermit 95、MS-DOS Kermit 以及 IBM Mainframe Kermit 为计算机间的通信提供了一种一致的、几乎通用的解决方案。

C-Kermit 8.0 版权 (C) 1985, 2001。纽约市哥伦比亚大学理事会版权所有。要了解有关使用和重新分发本软件的权利规定，请参阅 C-Kermit 的 COPYING.TXT 文件或使用 C-Kermit 的 COPYRIGHT 命令（摘要：自己使用本软件时不需要获得许可；使用源代码开放操作系统进行分发时也不需要许可；但对于某些其他形式的重新分发则需要获得许可）。

C-Kermit 8.0 由 HP公司（哥伦比亚大学的 Kermit 项目合作伙伴）引入到 HP-UX 中。

在 Frank da Cruz 和 Christine M. Gianone 编著的《Using C-Kermit》（Digital Press， 1997 年第二版）一书中全面讲述了 C-Kermit 6.0；请参阅本手册页末尾的“参考资料”部分。本手册页并不能替代该著作。如果您是热衷 C-Kermit 的用户，尤其是如果要编写 C-Kermit 脚本程序，则应该购买该手册。书籍销售是非赢利性 Kermit 项目的主要资金来源。

在第三版完成之前，在本书最新版出版后所添加的任何新功能都将记录在 *ckermit2.upd* 在线文档中。相关的提示、技巧、限制、约束将在 *ckcker.txt* （通用 C-Kermit）和 *ckuker.bwr* （UNIX 专用）中列出；请参阅下面的“文件”。在提出问题或请求技术支持之前请先查阅这些参考资料。

Kermit 软件适用于哥伦比亚大学的上百种计算机和操作系统。为了得到最佳的文件传输结果，请在其他计算机上使用 C-Kermit 时配套使用真正的哥伦比亚大学 Kermit 软件，如用于 Windows 95 和 NT 的 Kermit 95 或用于 DOS 3.x 或 Windows 的 MS-DOS Kermit。请参阅下面的“联系方式”。

运行模式

C-Kermit 可以在两种“模式”下使用：远程模式和本地模式。在 “远程模式”中，将从台式计算机连接到 HP-UX 系统，并在台式计算机和 HP-UX C-Kermit 之间进行文件传输。在该模式中，连接的建立（拨号、TELNET

连接等）是由台式计算机上的 Kermit 程序处理的。

在“本地模式”中，C-Kermit 通过直接串行连接、调制解调器拨号或网络连接与另一台计算机建立连接。用于本地模式时，C-Kermit 使用您的实际终端、仿真器、或 UNIX 工作站的终端窗口、或用于特定终端仿真的控制台驱动程序，建立终端到远程计算机的连接。

C-Kermit 还具有两种命令类型：常见的 UNIX 形式的命令行选项，及给出提示的交互式对话。“命令行选项”使您能够访问一个较小但非常有用的 C-Kermit 功能子集，以进行终端连接和文件传输，它还可以通过管道将文件传入或传出 Kermit 来进行传输。

“交互式命令”使您能够访问拨号、脚本编程、字符集转换，以及在通常情况下实现对所有的 C-Kermit 的功能的详细控制、显示及自动化。还可以将交互式命令汇集到命令文件或宏中。C-Kermit 的命令和脚本语言对于许多不同平台是可移植的。

启动 C-KERMIT

您可以通过键入 **/usr/bin/kermit** 或者只输入 **kermit** （如果您的路径中已包括 **/usr/bin** ）来启动 C-Kermit，可能需要在其后输入某些命令行选项。如果命令行中没有“操作选项”（在下文中解释），则 C-Kermit 会以交互式命令模式启动；这时您会看到一条问候消息，然后是“C-Kermit>”提示。如果命令行中确实包含操作选项，则 C-Kermit 会按指定的选项进行操作，然后直接退出并返回到 UNIX 中。在这两种方式下，C-Kermit 都会在执行任何其他命令之前先执行其初始化文件 **/usr/share/lib/kermit/ckermit.ini** 中的命令，除非在命令行中指定了“ **-Y** ”（大写）选项（表示将跳过初始化文件），或者命令行中包括了“ **-y** *filename* ” 选项（以指定其他初始化文件）。

文件传输

以下是进行 Kermit 文件传输的最常用方案。也有许多其他方法，而且大多数方法更为方便，但是这种基本方法可以在各种情况下使用。

- 在您的本地计算机上启动 Kermit 并建立到远程计算机的连接。如果 C-Kermit 在您的本地计算机上，则通过拨号连接时，可使用序列 SET MODEM TYPE *modem-name* 、 SET LINE *device-name* 、 SET SPEED *bits-per-second* 、及 DIAL *phone-number* ；进行直接连接时可使用 SET LINE 和 SPEED；通过网络连接时可使用 SET NETWORK *network-type* 和 SET HOST *host-name-or-address* 。
- 设置其他任何必需的通信参数，例如 PARITY、DUPLEX、和 FLOW-CONTROL。
- 执行 CONNECT 命令。
- 登录到远程计算机。
- 在远程计算机上启动 Kermit，并给出与文件、通信、或协议相关的参数所需的任何 SET 命令。如果将要传输二进制文件，则可以在要发送二进制文件的 Kermit 程序中输入 SET FILE TYPE BINARY 命令。
- 下载文件或文件组时，可在远程 Kermit 中输入 SEND 命令，并在后面输入文件名或使用文件“通配符”，例如：

  send oofa.txt   # (send one file)

send oofa.*   # (send a group of files)

上载一个或多个文件时，可在远程 Kermit 中输入 RECEIVE 命令。正在执行发送命令的 Kermit 会通知正在执行接收命令的 Kermit 每个文件的名称（及其他属性）。

- 退回到本地（台式）计算机的 Kermit 程序。如果在您的本地计算机上正在运行 C-Kermit，则键入 Ctrl-\c （Control - 反斜杠，然后是字母“c”；在 NeXT 工作站上应键入 Ctrl-] c）。如果运行的是 MS-DOS 或 Kermit 95，则使用 Alt-x（按住 Alt 键，再按“x”）。现在您应该可以看到本地 Kermit 程序的提示。
- 如果将要传输二进制文件，则可以在要发送这些文件的 Kermit 程序中输入 SET FILE TYPE BINARY 命令。
- 如果您将要 下载文件，则可以通知本地 Kermit 程序执行 RECEIVE 命令。如果您要 “上载”文件，则可以使本地 Kermit 程序执行 SEND 命令，并指定文件名或文件通配符形式。换句话说，先通知 远程 Kermit 程序所执行的命令：SEND 或 RECEIVE，然后退回到 本地 Kermit 中，并通知其执行相反的命令，即 RECEIVE 或 SEND。
- 当传输完成时，输入 CONNECT 命令。这样您又可以与远程计算机上的 Kermit 进行通信了。键入 EXIT 以回到远程计算机上的命令提示状态。当您完成远程计算机的使用时，可以注销，然后（如果必要的话）退回到本地计算机中的 Kermit。之后，您可以在本地 Kermit 程序中再次进行连接或退出该程序。

请注意，可以使用其他方法来简化文件传输的过程：“客户端/服务器操作”，在此方法中，所有命令都将在客户端中给出，并被自动传送给服务器；“自动下载”（及上载），在此方法中，远程 Kermit 将通过您的仿真终端自动启动文件传输。

在 C-Kermit 8.0 中设置了缺省文件传输协议，与以前的版本不同，后者假定传输时的连接通常是可靠的（使用具有硬件流量控制的 TCP/IP 和（或）错误更正式调制解调器），因而更注重于速度而不是稳定性。如果您遇到文件传输失败的情况，可使用 CAUTIOUS 命令或 ROBUST 命令来选择更加保守的（因而速度更慢）协议设置。要进行细微的性能调整，您可以选择特定数据包长度、窗口大小和作为策略前缀的控制字符，有关说明请参阅手册《Using C-Kermit》的第 12 章。

如果您要通过 Telnet 或其他可保证端对端可靠性的连接方式访问一台装有 C-Kermit 的远程主机，并且两个 Kermit 均支持这种连接（如 C-Kermit 8.0），则将自动使用 Kermit 协议的新“流”格式来提供接近 ftp 的速率（限制因素是源自远程 Telnet 或 Rlogin 服务器和（或）PTY 驱动程序的额外开销）。

其他功能

C-Kermit 包括的功能非常多，不可能在一个联机帮助页中全部说明。有关下列功能的详细信息，请参阅《Using C-Kermit》：建立连接、调制解调器拨号、网络和终端连接、键值映射、日志记录、文件传输选项及功能、疑难解答、客户端/服务器操作、终端连接和文件传输过程中的字符集转换、原始文件的上载和下载、宏构成、脚本编程、便捷功能、快捷方式，以及大量的表、示例和插图。

获得帮助

C-Kermit 有详实的内置帮助。通过在 C-Kermit> 提示符后键入 **?**，可以了解都存在哪些命令。您可以在 C-Kermit> 提示符后键入 HELP 来获得“快速入门”信息，或者键入 HELP 以及一个特定命令名来得到有关该命令的信息，例如：

help send

help set file

您可以在某命令的任意位置键入一个 ? 来获得关于当前命令字段的简要帮助。也可以键入 INTRO 命令获得关于 C-Kermit 的简要介绍，还可以键入 MANUAL 命令来访问该（或另一个）联机帮助页。最后，可以使用 SUPPORT 命令来查阅关于如何获得技术支持的说明。

命令的输入

对于交互模式命令，可以使用大写或小写字母，但要记住， UNIX 文件名是区分大小写的。只要缩写词只代表一种可能的含义时，就可以对命令进行缩写。键入命令时，可以使用下列编辑字符：

可使用删除、退格或 Rubout 键来清除最右边的字符。
使用 Ctrl-W 删除最右边的“单词”。
使用 Ctrl-U 删除当前命令行。
使用 Ctrl-R 重新显示当前命令。
使用 Ctrl-P 重新调用以前的命令（在命令缓冲区中回滚）。
使用 Ctrl-N 可在已回滚的命令缓冲区中向前滚动。
使用 Ctrl-C 取消当前命令。
使用 Tab、Esc 或 Ctrl-I 结束当前的关键字或文件名。
? 给出关于当前字段的帮助。

要进入命令并使之执行，可以按 Return 键或 Enter 键。

反斜杠符号

在交互式命令中， **** 字符（反斜杠）是一个前缀，用于输入特殊字符，包括不加此前缀时为非法字符的普通字符。在行末， **** 或 **-** （短划线）使下一行与当前行连续。除此以外， **** 后面的字符可以识别特殊字符的意义：

%	由用户定义的简单（标量）变量，例如 \%a 或 \%1
&	数组引用，例如 \&a[3]
$	环境变量，例如 \$(TERM)
v (or V)	内置变量，例如 \v(time)
f (or F)	函数，例如 \Fsubstring(\%a,3,2)
s (or S)	压缩子字符串表示形式、宏名称，例如 \s(foo[3:12])
:	压缩子字符串表示形式、所有变量，例如 \:(a[3:12])
d (or D)	基数为 10 的十进制数（1 到 3 位，0..255），例如 \d27
o (or O)	基数为 8 的八进制数（1 到 3 位，0..377），例如 \o33
x (or X)	基数为 16 的十六进制数（2 位，00..ff），例如 \x1b

\	反斜杠字符本身
b (or B)	“中断”信号（仅用于输出命令）
l (or L)	长“中断”信号（仅用于输出）
n (or N)	空 (0) 字符（仅用于输出）
a decimal digit	一个一位、两位或三位的十进制数字，例如 \27
{}	用于分组，例如 \{27}123
anything else:	按字面意思解释的后续字符。

请注意，这些数字被转换为具有该二进制编码 (0-255) 的字符，所以可以使用 **\7** 来表示时钟，使用 **\13** 来表示回车，使用 **\10** 表示换行。例如，要让 C-Kermit 将“时钟”发送到屏幕上，可以键入：

echo \7

命令列表

对于初学者非常重要的最常用命令使用了“*”进行标记：

程序管理

BACK	返回上一个目录。
BROWSE	调用 Web 浏览器。
* CD	更改目录
CHMOD	将给定文件的权限更改为给定的代码，该代码必须是一个八进制数，例如 664 或 775
PWD	输出工作目录。
GREP	在指定的一个或多个文件中搜索给定的字符串或模式。
CHECK	查看是否已配置了给定功能。
CLOSE	关闭一个连接或日志或其他本地文件。
COMMENT	引入一个整行注释。
COPYRIGHT	显示版权通知。
DATE	显示日期和时间。
* EXIT	退出程序，返回到 UNIX。
* HELP	显示给定命令的帮助信息。
* INTRO	输出 C-Kermit 的简短介绍。
KERMIT	根据提示给出命令行选项。
LOG	打开一个日志文件 -- 调试、打包、会话、事务处理。
PUSH	调用本地系统的交互式命令解释程序。
QUIT	与 EXIT 同义。
REDO	重新执行上一个命令。
RUN	运行程序或系统命令。
SET COMMAND	与命令相关的参数：字节大小、重新调用缓冲区。
SET PROMPT	C-Kermit 程序的交互式命令提示。

SET EXIT	与 C-Kermit 的退出或“SET LINE/HOST”操作相关的项目。
SHOW EXIT	显示 SET EXIT 的参数。
SHOW FEATURES	显示与 C-Kermit 一起构建的功能。
SHOW VERSIONS	显示每个源模块的版本号。
SUPPORT	了解如何获得技术支持。
SUSPEND	Suspend Kermit（仅在 Shell 支持作业控制时使用！）。
* SHOW	显示 SET 参数的值。
* TAKE	执行文件中的命令。
VERSION	显示 C-Kermit 程序版本号。
Z	与 SUSPEND 同义。
* Ctrl-C	中断一个正在执行的 C-Kermit 命令。
Ctrl-Z	与 SUSPEND 同义。
; 或 #	引入一个整行或尾部注释。
! 或 @	与 RUN 同义。
<	与 REDIRECT 同义。

连接的建立与释放:

* DIAL	拨打电话号码。
PDIAL	拨打一个电话号码的一部分。
* LOOKUP	查询电话号码并测试拨号规则。
ANSWER	等待电话并准备答复。
* HANGUP	挂断电话或者中断网络连接。
EIGHTBIT	将所有输入/输出都设置为 8 位的快捷方式。
PAD	X.25 PAD 的命令（仅用于 SunOS / Solaris / VOS）。
PING	检查远程 TCP/IP 主机的状态。
REDIAL	再次拨打最近刚拨打过的电话号码。
LOG CONNECTIONS	保存每个连接的记录。
REDIRECT	将命令的标准输入/输出重定向到通信连接。
PIPE	通过外部命令或程序建立连接。
SET CARRIER	终端连接上的载波处理。
* SET DIAL	与调制解调器拨号相关的参数。
* SET FLOW	通信线路流控制：AUTO、RTS/CTS、XON/XOFF 等。
* SET HOST	指定远程网络主机名或地址。
* SET LINE	指定串行通信设备名称，例如/dev/cul0p0。
SET PORT	与 SET LINE 同义。
* SET MODEM TYPE	指定在 SET LINE 设备上调制解调器的类型，例如 USR。
* SET NETWORK	网络类型、X.25（仅用于 SunOS / Solaris / VOS）或 TCP/IP。
SET TCP	指定 TCP 协议选项（高级）。
SET TELNET	指定 TELNET 协议选项。

SET X.25	指定 X.25 连接参数（仅用于 SunOS / Solaris / VOS）。
SET PAD	X.25 X.3 PAD 参数（仅用于 SunOS / Solaris / VOS）。
* SET PARITY	为通信设置字符奇偶校验（无、偶等）。
* SET SPEED	串行通信设备的速率，例如 2400、9600、57600。
SET SERIAL	设置串行通信的数据大小、奇偶校验、停止位。
SET STOP-BITS	设置串行通信停止位。
SHOW COMM	显示所有的通信设置。
SHOW CONN	显示关于当前连接的信息。
SHOW DIAL	显示 SET DIAL 的值。
SHOW MODEM	显示调制解调器类型、信号等。
SHOW NETWORK	显示与网络相关的项目。
* TELNET	= SET NETWORK TCP/IP、SET HOST ...、CONNECT。
RLOGIN	建立 RLOGIN 连接（需要权限）。
TELOPT	发送 TELNET 选项协定（高级）。
CLOSE	关闭当前连接。

终端连接

* C	CONNECT 的特定缩写。
* CONNECT	建立到远程计算机的终端连接。
LOG SESSION	记录终端会话。
SET COMMAND	C-Kermit 和您的键盘与屏幕之间的字节大小。
* SET DUPLEX	指定在执行 CONNECT 期间由哪一方回应。
SET ESCAPE	执行 CONNECT 期间“转义命令”的前缀。
SET KEY	在“连接”模式下重新定义键值。
SET TERMINAL	终端连接项目：字节大小、字符集、回应等。
SHOW ESCAPE	显示当前“连接”模式下的转义字符。
SHOW KEY	显示键编码和指定的值或宏。
SHOW TERMINAL	显示 SET TERMINAL 项目。
* Ctrl-\	“连接”模式下的转义字符，后面是其他字符： C 表示返回到 C-Kermit> 提示。 B 表示发送“中断”信号。 ? 表示请参阅其他选项。

文件传输

ADD SEND-LIST	将文件规格说明添加到“发送列表”中。
ADD BINARY-PATTERNS	将模式添加到二进制文件模式列表中。
ADD TEXT-PATTERNS	将模式添加到文本文件模式列表中。
ASSOCIATE	带有传输字符集的文件字符集。
LOG SESSION	在不进行错误检查的情况下下载文件。

* SEND	发送一个或多个文件。
MSEND	多次执行 SEND - 接收由空格分隔的文件的列表。
MOVE	通过 SEND 发送源文件，如果发送成功则删除它。
MMOVE	多次执行 MOVE - 接收由空格分隔的文件的列表。
MAIL	像发送电子邮件一样将文件通过 SEND 发送到其他 Kermit 中。
RESEND	继续进行未完成的 SEND 命令。
PSEND	发送文件的部分内容。
* RECEIVE	被动地等待从其他 Kermit 收到文件。
* R	RECEIVE 的专用缩写。
* S	SEND 的专用缩写。
GET	要求服务器发送指定的一个或多个文件。
MGET	类似于 GET，但接收文件列表。
REGET	继续未完成的从服务器下载的任务。
G	GET 的特定缩写。
FAST	快速文件传输设置的快捷方式。
CAUTIOUS	中速文件传输设置的快捷方式。
ROBUST	保守文件传输设置的快捷方式。
SET ATTRIB	控制文件属性的传输。
* SET BLOCK	选择错误检查级别：1、2、或 3。
SET BUFFERS	发送和接收数据包的缓冲区大小。
SET PREFIX	在文件传输中将哪些控制字符“解除前缀”。
SET DELAY	发送第一个数据包前的等待时间。
SET DESTINATION	将收到的文件发送到磁盘、打印机或屏幕。
* SET FILE	传输模式（类型）、字符集、冲突操作等。
* SET RECEIVE	入站数据包的参数：包长度等。
SET REPEAT	重复计数压缩参数。
SET RETRY	数据包重发的限制次数。
SET SEND	出站数据包的参数：长度等。
SET HANDSHAKE	通信线路上半双工数据包的转向字符。
SET LANGUAGE	启用语言特定的字符集转换。
PATTERNS	关闭基于文件名模式的文本/二进制模式转换。
SET SESSION-LOG	会话日志的文件类型：文本或二进制。
SET TRANSFER	文件传输参数：字符集、显示等。
SET TRANSMIT	控制 TRANSMIT 命令的执行。
SET UNKNOWN	指定对未知字符集的处理。
* SET WINDOW	文件传输包窗口大小：1-31。
SHOW ATTRIB	显示 SET ATTRIBUTE 的值。
SHOW CONTROL	显示控制字符前缀表。

* SHOW FILE	显示与文件相关的设置。
SHOW PROTOCOL	显示与协议相关的设置。
SHOW LANGUAGE	显示与语言相关的设置。
SHOW TRANSMIT	显示 SET TRANSMIT 的值。
* STATISTICS	显示时间最近的文件传输的统计。
TRANSMIT	发送文件，但不进行错误检查。
XMIT	与 TRANSMIT 同义。

SEND 命令选项。

/AS-NAME:	所发送文件的名称。
/AFTER:	发送在某个日期-时间后修改的文件。
/BEFORE:	发送在某个日期-时间之前修改的文件。
/BINARY	以二进制方式发送。
/COMMAND	从某个命令的标准输出发送。
/DELETE	成功发送文件后将其删除。
/EXCEPT:	不发送文件名与给定模型相匹配的文件。
/FILTER:	通过给定的过滤程序传送文件内容。
/FILENAMES:	指定如何发送文件名。
/LARGER-THAN:	发送大于给定大小的文件。
/LIST:	发送给定文件中列出其文件名的文件。
/MAIL:	将文件以电子邮件方式发送到给定地址。
/MOVE-TO:	成功发送文件后将其移动到指定目录。
/NOT-AFTER:	发送修改时间不晚于给定日期-时间的文件。
/NOT-BEFORE:	发送修改时间不早于给定日期-时间的文件。
/PATHNAMES:	指定如何发送路径名。
/PRINT:	发送要输出的文件。
/PROTOCOL:	使用给定协议发送文件。
/QUIET	不显示文件传输进度。
/RECOVER	从失败点恢复中断的传输。
/RECURSIVE	发送目录树。
/RENAME-TO:	发送成功文件后以指定文件名重命名该文件。
/SMALLER-THAN:	发送小于给定大小的文件。
/STARTING-AT:	发送以给定字节号开头的文件。
/SUBJECT:	SEND/MAIL 的主题。
/TEXT	以文本模式发送。

GET 和 **RECEIVE** 命令选项

/AS-NAME:	以给定的名称存储传入的文件。
/BINARY	如果未指定传输模式，则以二进制模式接收。

/COMMAND:	将传入的文件数据发送至给定命令。
/EXCEPT:	不接受名称匹配的传入文件。
/FILENAMES:	如何处理传入文件的名称。
/FILTER:	传入文件数据的过滤程序。
/MOVE-TO:	成功接收文件后将其移动到何处。
/PATHNAMES:	如何处理传入的路径名。
/PROTOCOL:	接收协议（仅用于 RECEIVE 命令）。
/RENAME-TO:	成功接收文件后为其指定新的名称。
/QUIET:	禁止文件传输显示。
/TEXT	未指定传输模式时按文本模式接收。

仅用于 **GET** 的选项

/DELETE	通知服务器在成功传输每个文件后将其删除。
/RECOVER	从失败点恢复中断的文件传输。
/RECURSIVE	通知服务器发送目录树。

文件管理

* CD	更改目录。
* PWD	显示当前工作目录。
COPY	复制一个文件。
* DELETE	删除一个或多个文件。
* DIRECTORY	显示一个目录列表。
EDIT	编辑一个文件。
MKDIR	创建目录。
PRINT	在本地打印机上打印本地文件。
PURGE	删除备份文件。
RENAME	更改一个本地文件的名称。
RMDIR	删除一个目录。
SET ROOT	为文件指定超级用户权限，使其可以访问给定目录，禁止访问系统、Shell 命令和外部程序。
SET PRINTER	选择打印机设备。
SPACE	显示当前磁盘空间使用状态。
SHOW CHARACTER-SETS	显示字符集转换信息。
TRANSLATE	转换本地文件的字符集。
TYPE	将文件内容显示到屏幕。
TYPE /PAGE	将文件内容显示到屏幕，每次满屏时暂停。
XLATE	与 TRANSLATE 同义。

客户端/服务器操作

BYE	终止远程 Kermit 服务器，并注销其作业。

DISABLE	服务器运行期间不允许访问所选功能。
E-PACKET	发送含错误信息的数据包。
ENABLE	服务器运行期间允许访问所选功能。
FINISH	命令远程 Kermit 服务器退出，但不注销。
G	GET 的专用缩写。
GET	从远程 Kermit 服务器中获取文件。
QUERY	（与 REMOTE QUERY 相同）
RETRIEVE	类似于 GET，但服务器在命令完成后删除文件。
REMOTE xxx	服务器命令，可以用 > 或	重定向。
REMOTE ASSIGN	(RASG) 为变量赋值。
REMOTE CD	(RCD) 通知 Kermit 服务器更改其目录。
REMOTE COPY	(RCOPY) 通知服务器复制文件。
REMOTE DELETE	(RDEL) 通知服务器删除文件。
REMOTE DIR	(RDIR) 向服务器请求目录列表。
REMOTE EXIT	(REXIT) 请求服务器程序退出。
REMOTE HELP	(RHELP) 请求服务器发送帮助消息。
REMOTE HOST	(RHOST) 请求服务器向其主机发送执行命令的请求。
REMOTE KERMIT	(RKER) 向服务器发送一个交互的 Kermit 命令。
REMOTE LOGIN	针对远程 Kermit 服务器对您自己进行身份验证。
REMOTE LOGOUT	从之前使用 LOGIN 登录的 Kermit 服务器注销。
REMOTE MKDIR	(RMKDIR) 通知服务器创建目录。
REMOTE PRINT	(RPRINT) 在服务器打印机上打印本地文件。
REMOTE PWD	(RPWD) 请求服务器显示其当前（工作）目录。
REMOTE QUERY	(RQUERY) 获取变量值。
REMOTE RENAME	(RRENAME) 通知服务器重命名文件。
REMOTE RMDIR	(RRMDIR) 通知服务器删除目录。
REMOTE SET	向远程服务器发送一个 SET 命令。
REMOTE SPACE	查询服务器剩余磁盘空间。
REMOTE TYPE	请求服务器在屏幕上显示文件内容。
REMOTE WHO	查询服务器的“who”或“finger”列表。
SERVER	指定 Kermit 服务器。
SET SERVER	服务器运行参数。
SHOW SERVER	显示 SET SERVER、ENABLE/DISABLE 项目。

脚本编程

ASK	提示用户，并将用户的回答存储到一个变量中。
ASKQ	类似于 ASK，但不在屏幕中回显（对于口令很有用）。
ASSERT	评估状态并相应地设置 SUCCESS/FAILURE。
ASSIGN	将已求值的字符串赋给变量或宏。

CLEAR	清除通信设备输入缓冲区或其他项目。
CLOSE	关闭连接、日志或其他文件。
DECLARE	声明数组。
DECREMENT	变量减 1（或减其他数）。
DEFINE	定义变量或宏。
DO	执行宏（“DO”可以忽略）。
ECHO	在屏幕上显示文本。
ELSE	与 IF 成对使用。
END	一个命令文件或宏。
EVALUATE	算术表达式。
FAIL	设置 FAILURE。
FOPEN	打开本地文件。
FREAD	读取使用 FOPEN 打开的文件。
FWRITE	写入使用 FOPEN 打开的文件。
FSEEK	在使用 FOPEN 打开的文件中寻找指定的位置。
FCLOSE	关闭使用 FOPEN 打开的文件。
FOR	以计数循环结构重复执行命令。
FORWARD	仅向前跳转 (GOTO)。
GETC	发出提示，并从键盘获取一个字符。
GETOK	提出问题，得到“是”或“否”的回答，并设置 SUCCESS 或 FAILURE。
GOTO	转到一个命令文件或宏中有标记的命令。
IF	根据条件执行接下来的命令。
INCREMENT	使变量加 1（或加其他数）。
INPUT	根据给定文本匹配来自另一台计算机的字符。
LOCAL	在宏中声明局部变量。
MINPUT	类似于 INPUT，但是允许有几个匹配字符串。
MSLEEP	休眠指定的毫秒数。
OPEN	打开一个本地文件进行读或写。
OUTPUT	将文本发送到另一台计算机中。
O	OUTPUT 的专用缩写。
PAUSE	在指定的秒数内暂停任何操作。
READ	从一个本地文件中读取一行并将其赋给变量。
REINPUT	重新检查以前从另一台计算机收到的文本。
RETURN	从用户定义的函数中返回。
SCREEN	屏幕的操作 - 清除、位置指针等。
SCRIPT	执行 UUCP 格式的登录脚本。
SET ALARM	设置将与 IF ALARM 一起使用的定时器；并通过 SHOW ALARM 来显示。
SET CASE	设置在比较字符串时如何处理字母大小写。

SET COMMAND	QUOTING 打开/关闭反斜杠符号的转义功能。
SET COUNT	用于计数式循环结构。
SET INPUT	控制 INPUT 命令的行为。
SET MACRO	控制宏的执行。
SET TAKE	控制 TAKE 文件的执行。
SHIFT	将宏的参数向左移位指定的位数。
SHOW ARGUMENTS	显示当前宏的参数。
SHOW ARRAYS	显示活动数组的信息。
SHOW COUNT	显示 COUNT 当前值。
SHOW FUNCTIONS	列出可用的 \f() 函数的名称。
SHOW GLOBALS	列出已定义的全局变量 \%a..\%z。
SHOW MACROS	列出一个或多个宏定义。
SHOW SCRIPTS	显示与脚本相关的设置。
SHOW VARIABLES	显示所有 \v() 变量的值。
SLEEP	休眠指定的秒数。
SORT	对数组排序（多个选项）。
STATUS	显示前一个命令的 SUCCESS 或 FAILURE。
STOP	停止执行宏或命令文件，返回到提示。
SUCCEED	设置 SUCCESS。
SWITCH	根据变量值执行选定的命令。
TAKE	执行文件中的命令。
UNDEFINE	取消一个变量的定义。
WAIT	等待指定的调制解调器信号。
WHILE	当条件为真时重复执行命令。
WRITE	向一个本地文件写入数据。
WRITE-LINE	向一个本地文件写入一行（记录）。
WRITELN	与 WRITE-LINE 同义。
XECHO	类似于 ECHO，但结尾处没有 CRLF。
XIF	扩展的 IF 命令。

内置变量

内置变量的格式为 \v（名称），可用于任何命令中，通常用于脚本编程。不能更改内置变量。键入 SHOW VARIABLES 可得到当前变量列表。

\v(argc)	当前宏中的参数个数
\v(args)	程序命令行的参数个数
\v(blockcheck)	当前 SET BLOCK-CHECK 的类型
\v(browser)	当前 Web 浏览器
\v(browsopts)	当前 Web 浏览器选项

\v(browsurl)	最近访问过的 Web 浏览器站点 (URL)
\v(byteorder)	硬件字节顺序
\v(charset)	当前文件字符集
\v(cmdbufsize)	命令缓冲区大小
\v(cmdfile)	当前命令文件的名称（如果存在的话）
\v(cmdlevel)	当前命令级别
\v(cmdsource)	当前的命令来源：宏、文件等。
\v(cols)	屏幕列数
\v(connection)	连接类型：串行连接、TCP/IP 等。
\v(count)	当前 COUNT 值
\v(cps)	最近发生的文件传输的速率，用每秒字符数表示。
\v(cpu)	编译 C-Kermit 所面向的 CPU 类型
\v(crc16)	最近的文件传输的 16 位 CRC 校验
\v(ctty)	控制终端的设备名称
\v(d$ac)	设置 DIAL AREA-CODE 的值
\v(d$cc)	设置 DIAL COUNTRY-CODE 的值
\v(d$ip)	设置 DIAL INTL-PREFIX 的值
\v(d$lc)	设置 DIAL LD-PREFIX 的值
\v(d$px)	设置 DIAL PBX-EXCHANGE 的值
\v(date)	日期（如 8 Feb 1993）
\v(day)	星期几
\v(dialcount)	DIAL 重试计数器的当前值
\v(dialnumber)	最近拨打过的电话号码
\v(dialresult)	来自调制解调器的最近拨号的结果消息或编码
\v(dialstatus)	DIAL 命令的返回代码（0 = 正常，22 = 设备忙，等等）
\v(dialsuffix)	当前 SET DIAL SUFFIX 的值
\v(dialtype)	最近拨打的电话类型代码
\v(directory)	当前/缺省目录
\v(download)	当前下载目录（如果有的话）
\v(editor)	首选编辑器
\v(editfile)	最近编辑过的文件
\v(editopts)	编辑器选项
\v(errno)	当前“errno”（系统错误号）的值
\v(errstring)	与系统错误号相关联的错误消息字符串
\v(escape)	“连接”模式下转义符号的十进制 ASCII 码值
\v(evaluate)	最近执行的 EVALUATE 命令的结果
\v(exitstatus)	当前退出状态（0 = 正常，非 0 = 存在错误）
\v(filename)	当前传输的文件的名称

\v(filenumber)	当前传输的文件的编号（1 = 第一个，等等）
\v(filespec)	最近执行的 SEND/RECEIVE/GET 命令中给定的文件规格
\v(fsize)	最近传输的文件的大小
\v(ftype)	SET FILE TYPE 的值（text、binary）
\v(herald)	C-Kermit 的程序使者
\v(home)	主目录
\v(host)	计算机主机名（运行 C-Kermit 的计算机）
\v(hwparity)	SET PARITY HARDWARE 设置（如果有的话）
\v(input)	当前 INPUT 缓冲区的内容
\v(inchar)	最近通过 INPUT 输入的字符
\v(incount)	在上一个 INPUT 期间到达的字符数
\v(inidir)	找到初始化文件的目录
\v(inmatch)	与给定的 \fpattern() 相匹配的 [M]INPUT 内容。
\v(instatus)	最近的 INPUT 命令的状态
\v(intime)	最近的 INPUT 成功执行所用的时间（毫秒）
\v(inwait)	最近的 [M]INPUT 的时间限制
\v(ipaddress)	C-Kermit 的计算机的 IP 地址（如果已知）
\v(kbchar)	中断 PAUSE、INPUT 等命令的键盘字符。
\v(line)	由 LINE 或 HOST 设置的当前通信设备
\v(local)	若在远程模式下则为 0，在本地模式下为 1
\v(lockdir)	本平台上 UUCP 锁文件的目录
\v(lockpid)	当端口在用时，在锁文件中找到的进程 ID
\v(maclevel)	当前宏堆栈层
\v(macro)	当前执行的宏的名称（如果有的话）
\v(math_e)	浮点数常量 e
\v(math_pi)	浮点数常量 pi
\v(math_precision)	浮点数精度（位数）
\v(minput)	最近的 MINPUT 命令的结果
\v(model)	计算机硬件模型（如果已知）
\v(modem)	当前调制解调器类型
\v(m_aa_off)	用于关闭自动应答的调制解调器命令
\v(m_aa_on)	用于打开自动应答的调制解调器命令
\v(m_xxxxx)	（很多其他调制解调器命令）
\v(m_sig_xx)	调制解调器信号 xx 的值
\v(name)	调用 C-Kermit 时使用的名称（kermit、wermit 等）
\v(ndate)	当前日期，如 19930208 (yyyymmdd)
\v(nday)	用数字表示星期几（0 = 星期天）
\v(newline)	独立于系统的换行字符或序列

\v(ntime)	当前本地时间距午夜时间的秒数（中午 = 43200）
\v(osname)	操作系统名称
\v(osrelease)	操作系统发行版
\v(osversion)	操作系统版本
\v(packetlen)	当前 SET RECEIVE PACKET-LENGTH 的值
\v(parity)	当前的奇偶校验设置
\v(pexitstat)	最近派生的进程的退出状态
\v(pid)	C-Kermit 的进程 ID
\v(platform)	特定的计算机和（或）操作系统
\v(program)	该程序的名称（“C-Kermit”）
\v(protocol)	当前选定的文件传输协议
\v(p_8bit)	当前第 8 位前缀（Kermit 协议）
\v(p_ctl)	当前控制字符前缀（Kermit 协议）
\v(p_rpt)	当前重复计数前缀（Kermit 协议）
\v(query)	最新的 REMOTE QUERY 命令的结果
\v(return)	最新的 RETURN 值
\v(rows)	终端屏幕中的行数
\v(sendlist)	SEND-LIST 中的条目数
\v(serial)	采用 8N1 格式的串行端口设置
\v(speed)	当前速率（如果已知），否则为+“unknown”
\v(startup)	启动 C-Kermit 时的当前目录
\v(status)	0 或 1（上一个命令的“SUCCESS”或“FAILURE”）
\v(sysid)	C-Kermit 计算机的平台 ID 代码 (U1=UNIX)
\v(system)	UNIX（操作系统系列的名称）
\v(terminal)	终端类型
\v(test)	C-Kermit 的测试版本（如果有的话，例如，Beta.10）
\v(textdir)	C-Kermit 认为其文本文件所在的位置
\v(tfsize)	最近传输的文件组的总大小
\v(time)	时间，比如 13:45:23 (hh:mm:ss)
\v(tmpdir)	临时目录
\v(trigger)	从 CONNECT 中触发返回的最新字符串
\v(ttyfd)	当前通信设备的文件描述符
\v(ty_xx)	由 TYPE 在内部使用
\v(userid)	运行 C-Kermit 的用户的 ID
\v(version)	以数字表示的 Kermit 的版本，例如，501190。
\v(window)	当前窗口大小（SET WINDOW 的值）
\v(xferstatus)	最近发生的文件传输的状态
\v(xfermsg)	结束最近一次传输的错误消息（如果有的话）

\v(xfer_xxx)	来自最后一次文件传输的各种统计。
\v(xprogram)	C-Kermit
\v(xversion)	与 \v(version) 同义

内置功能

内置功能可以作为 \Fname(args) 调用，可以用于任何命令中，且通常用于脚本程序中。为当前列表键入 SHOW FUNCTIONS。为参数和返回值的描述键入“help function <name>”，例如 **help function basename** 。

命令行选项

C-Kermit 可以在命令行中接受经过时间检验的 UNIX 形式的命令（或“选项”）。字母的大小写是非常重要的。所有的选项都是可选的。如果命令行中包括一个或多个操作选项，则 Kermit 会在执行命令行选项后立即退出，否则会进入交互命令模式。

kermit [*filename*] [**-***x arg* [**-***x arg*]...[**-***yyy*]...]]

其中：

filename 是要执行的命令文件的名称，

-*x* 是需要参数的选项，

-*y* 是不带参数的选项。

操作

-s files	发送文件
-s -	发送文件来自标准输入的文件
-r	接收文件
-k	将文件接收到标准输出中
-x	进入服务器模式
-O	类似于 -x，但在一次事务处理后退出
-f	终止远程服务器
-g files	从服务器中获取远程文件（引用通配符）
-G files	类似于 -g，但将文件发送到标准输出中
-a name	备用文件名，与 -s、-r、-g 一起使用
-c	在文件传输前连接，与 -l 或 -j 一起使用
-n	在文件传输后连接，与 -l 或 -j 一起使用

设置

-l line	通信线路设备（用于串行连接）
-l n	通信设备的打开文件描述符
-j host	TCP/IP 网络主机名（用于网络连接）
-J host	以类似于 TELNET 的方式连接，当连接关闭时退出
-l n	TCP/IP 连接的打开文件描述符（n = 数字）

-X	X.25 网络地址
-Z	X.25 连接的打开文件描述符
-o n	X.25 关闭的用户组调用信息
-u	X.25 受话人付款的电话
-q	文件传输期间保持静默状态
-I	连接是可靠的（例如，TCP 或 X.25）
-8	8 位清除
-0	在连接模式下 100% 透明度（无转义）
-i	以二进制模式传输文件
-T	以文本模式传输文件
-P	发送/接收文本路径（文件）名
-b bps	串行线路速率，例如 1200
-m name	调制解调器类型，例如 hayes
-p x	奇偶校验，x = e、o、m、s 或 n
-t	半双工，xon 握手
-e n	接收包的长度
-v n	窗口大小
-L	与 -s 一起使用，以选择递归目录传输
-Q	快速文件传输的设置
-w	覆盖同名文件，但不备份旧文件
-D n	延迟 n 秒发送文件
-V	“手动模式” = SET FILE PATTERNS OFF， SET TRANSFER MODE MANUAL。

其他

-y name	备用的初始化文件名
-Y	跳过初始化文件
-R	建议 C-Kermit 只在远程模式下使用该命令。
-d	将调试信息记录到文件 debug.log 中
-S	在执行操作命令后停留，但不退出
-C "cmds"	以逗号分隔的交互模式命令
-z	强制在前台运行
-B	强制在后台（批）运行
-h	输出命令行选项帮助屏幕
=	忽略后面的所有文本
--	与 = 同义

命令行示例

远程模式示例（C-Kermit 在远端上）：

kermit -v 4 -i -s oofa.bin

该命令使用窗口大小 4 (-v 4) 以二进制模式 (-i) 发送文件 oofa.bin。

本地模式的示例（C-Kermit 建立连接）：

kermit -l /dev/tty0p0 -b 19200 -c -r -n

该命令通过 /dev/tty0p0 建立 19200 bps 直接连接，首先进行连接 (-c) 使您可以登录和启动远程 Kermit 程序并通知其发送文件，然后该程序会接收文件 (-r) 并进行反向连接 (-n)，这样您就可以完成任务并注销。

为了拨出，您必须指定调制解调器类型，而且可能必须使用一个不同的设备名：

kermit -m hayes -l /dev/cul0p0 -b 2400 -c -r -n

文件

$HOME/.mykermrc	您的个人 C-Kermit 定制文件。
$HOME/.kdd	您的个人拨号目录。
$HOME/.ksd	您的个人服务目录。
/usr/share/lib/kermit/READ.ME	HP-UX C-Kermit 的概述，请阅读
/usr/share/lib/kermit/COPYING.TXT	版权、许可、放弃声明
/usr/share/lib/kermit/ckermit.ini	系统级初始化文件
/usr/share/lib/kermit/ckermod.ini	自定义文件示例
/usr/share/lib/kermit/ckermit.kdd	拨号目录示例
/usr/share/lib/kermit/ckermit.ksd	服务目录示例
/usr/share/lib/kermit/ckermit2.txt	对《Using C-Kermit》第二版的更新
/usr/share/lib/kermit/ckcbwr.txt	C-Kermit 的“注意”文件 - 线索 & 提示
/usr/share/lib/kermit/ckubwr.txt	特定于 UNIX 的注意文件
/usr/share/lib/kermit/ck*.txt	其他无格式文本文件
/usr/share/lib/kermit/ckedemo.ksc	《Using C-Kermit》中的宏
/usr/share/lib/kermit/ckevt.ksc	同上
/usr/share/lib/kermit/ckepager.ksc	Alpha pager 脚本
/var/spool/locks/LCK..*	UUCP 锁文件

为了进行 “个性化定制”，可将文件 **/usr/share/lib/kermit/ckermod.ini** 复制到您的主目录中，进行任何所需的更改，然后将此文件重新命名为 **.mykermrc** 。

您也可以创建类似于 **/usr/share/lib/kermit/ckermit.kdd** 中的示例的个性化 “拨号目录”。您的个性化拨号目录应该存储为您主目录中的 **.kdd** ，以及您个人网络目录中的 **.knd** 。有关详细信息，请参阅《Using C-Kermit》的第五章和第六章。

您也可以创建类似于 **/usr/share/lib/kermit/ckermit.ksd** 中的示例的个性化 “服务目录”。您的个性化服务目录应该存储为您主目录中的 **.ksd** 。有关说明请参阅《Using C-Kermit》的第七章。

示例文件对 C-Kermit 的脚本程序结构进行了举例说明；它们将在此书的第 17 至 19 章进行讨论。您可以通过在 C-Kermit> 提示符后键入适当的 TAKE 命令来运行这些脚本程序，例如： **take /usr/share/lib/kermit/ckedemo.ini**.

作者

哥伦比亚大学的 Frank da Cruz 以及全世界的众多志愿程序员。请参阅《Using C-Kermit》中的 Acknowledgements 部分。

参考资料

Frank da Cruz and Christine M. Gianone,

《Using C-Kermit》, Second Edition, 1997 年，共 622 页， Digital Press / Butterworth-Heinemann, 225 Wildwood Street, Woburn, MA 01801, USA. ISBN 1-55558-164-1。（美国用户请拨打 +1 800 366-2665 订购 Digital Press 出版的书籍）也有由汉诺威大学的 Verlag Heinze Heise 所译的德语版。

Frank da Cruz,

《Kermit, A File Transfer Protocol》 Digital Press / Butterworth-Heinemann, Woburn, MA, USA (1987). ISBN 0-932376-88-6。 Kermit 文件传输协议规范。

Christine M. Gianone,

《Using MS-DOS Kermit》, Digital Press / Butterworth-Heinemann, Woburn, MA, USA (1992)。 ISBN 1-5558-082-3。 也有由 Heise 所译的德语版和凡尔赛的 Heinz Schiefer &Cie 所译的法语版。

《Kermit News》,

第四期 (1990) 和第五期 (1993)，哥伦比亚大学，是有关 Kermit 文件传输性能的详细讨论。

诊断信息

C-Kermit 自身生成的诊断信息通常是自描述性质的。除此以外，每个命令都返回一个成功或失败状态，此状态可用 IF FAILURE 命令或 IF SUCCESS 命令来测试。此外，程序本身还会在操作成功时返回退出状态代码零，在发生任何失败操作时返回非零退出状态代码。

缺陷

有关 Usenet 的讨论，请参阅 comp.protocols.kermit.* newsgroups，或者参阅文件 ckcker.bwr 和 ckuker.bwr 以获得缺陷列表、提示、技巧等。可通过电子邮件将缺陷报告给 **kermit-support@columbia.edu** 。有关技术支持的详细信息，请访问网站 **http://www.columbia.edu/kermit/support.html** 。

联系方式

有关 Kermit 软件和文档的更多信息，请访问 Kermit Web 网站:

http://www.columbia.edu/kermit/

或写信到:

The Kermit Project
Columbia University
612 West 115th Street
New York, NY 10025-7221
USA

或发送电子邮件到 **kermit@columbia.edu** 。或拨打 +1 212 854-3703。 或发传真到 +1 212 663-8202。

名称

keylogin - 使用 keyserv 解密并存储私用密钥

概要

/usr/bin/keylogin [**-r**]

说明

keylogin 命令会提示用户输入口令，并使用该口令解密用户的私用密钥。密钥可能位于 **/etc/publickey** 文件中（请参阅 *publickey*(4)）中，或用户主域中的 NIS 映射“publickey.byname”或 NIS+ 表“cred.org_dir”中。在 **/etc/nsswitch.conf** 文件（请参阅 *nsswitch.conf*(4)）中指定了来源及其查找顺序。用户的私用密钥解密后，会由本地密钥服务器进程 *keyserv*(1M) 存储。当发出对任何安全 RPC 服务（如 NIS+）的请求时，会使用此存储密钥。程序 *keylogout*(1) 可用于删除由 **keyserv** 存储的密钥。

如果 **keylogin** 无法获取调用者的密钥，或给出的口令不正确，则该命令将失败。对于新用户或新主机，可使用 *newkey*(1M)、 *nisaddcred*(1M) 或 *nisclient*(1M) 添加新密钥。

选项

-r 更新 **/etc/.rootkey** 文件。此文件包含超级用户的未加密私用密钥。只有超级用户才可以使用此选项。使用该选项是为了作为超级用户运行的进程可以发出已验证的请求，而无需管理员在系统启动时明确地作为超级用户运行 **keylogin** （请参阅 *keyserv*(1M)）。当 publickey 数据库中主机的条目已更改，并且 **/etc/.rootkey** 文件中关于 publickey 数据库中存储的实际密钥对信息已过期时，管理员应使用 **-r** 选项。**/etc/.rootkey** 文件的权限是，只有超级用户可以对其进行读取和写入，系统上的其他任何用户都没有读写权限。

作者

keylogin 由 Sun Microsystems, Inc. 开发。

文件

/etc/.rootkey 超级用户的私用密钥

另请参阅

chkey(1)、 keylogout(1)、 login(1)、 keyserv(1M)、 newkey(1M)、 nisaddcred(1M)、 nisclient(1M)、 publickey(4)、 nsswitch.conf(4)。

名称

keylogout - 删除由 keyserv 存储的私用密钥

概要

/usr/bin/keylogout [**-f**]

说明

keylogout 删除通过密钥服务器进程 *keyserv*(1M) 存储的密钥。这将撤消对该密钥的进一步访问；然而，当前会话密钥在过期或被刷新之前可以一直保持有效。

删除由 **keyserv** 存储的密钥将导致需要安全 RPC 服务的任何后台作业或安排的 *at*(1) 作业失败。由于在一台计算机上只保存密钥的一个副本，所以建议不要在您的 **.logout** 文件中调用该命令，因为这会影响同一台计算机上的其他会话。

选项

-f 强制 **keylogout** 删除超级用户的私用密钥。缺省情况下，不允许超级用户执行 **keylogout** ，因为这样会中断该超级用户启动的所有 RPC 服务。

作者

keylogout 由 Sun Microsystems, Inc. 开发。

另请参阅

at(1)、 chkey(1)、 login(1)、 keylogin(1)、 keyserv(1M)、 newkey(1M)、 publickey(4)。

名称

keysh - 环境相关功能键 Shell

概要

keysh

说明

keysh 是标准 Korn-Shell 的扩展（有关基本 Korn-Shell 功能的说明，请参阅 *ksh*(1)）。

keysh 使用分层功能键菜单和环境相关帮助来帮助用户构建命令行，将 Korn-Shell 的强大功能与简便易用的菜单系统组合在一起。

keysh 是完全由数据驱动的，因此可以根据需要方便地扩展其菜单和帮助。

请注意，在 **keysh** 调用过程中，环境变量 **$TERM** 必须指定 *terminfo*(4) 数据库中定义的终端类型（请参阅下面的环境变量）。

命令输入

keysh 持续解析命令行，并随时在功能键标签上向用户显示一组适当的 当前选项。

用户可以选择这些功能键，在命令行上创建可读的 功能键命令。 **keysh** 会自动将这些功能键命令转换为等效的 *HP-UX* 命令，然后再执行这些命令。

或者，用户可以完全忽略功能键，而选择直接输入传统的 HP-UX 命令，就像在使用 Korn-shell 一样。

在命令输入过程中， **keysh** 通常会在屏幕的底部显示一个 状态行。该状态行包含主机名、当前目录以及日期和时间等信息。

只要用户 必须执行操作来完成当前功能键命令， **keysh** 就会在状态行的位置暂时显示 提示消息。该消息会简述必需的操作。

功能键类型

keysh 提供了四种基本功能键类型：

--Help-- 如果选择 **--Help--** 功能键，可使 **keysh** 显示与下一个选定的功能键相关联的帮助信息，而不是实际执行其操作。

--More-- 如果当前选项多于功能键， **keysh** 就会将选项分为选项组，并随第一个选项组显示一个特殊的 **--More--** 功能键。通过选择 **--More--** 功能键，可使 **keysh** 按顺序显示下一组功能键，最后会循环回到第一个选项组。

<*param*> *parameter* 功能键显示为位于一对小于和大于符号之间的名称。它们指示应该在此处向命令行中输入用户提供的文本（如文件名），而不是实际选择该功能键。（如果实际选择该功能键，仅会使 **keysh** 在状态行上显示提示消息；命令行仍保持不变）。

option 其他所有功能键都是 *option* 功能键，它们可用于在命令行中插入相应的命令或选项。

功能键可以按从左到右的顺序选择。

编辑命令行

keysh 支持常规的 Korn-Shell 命令行编辑模式。此外， **keysh** 还识别 *terminfo*(4) 数据库中定义的大多数终端上的光标移动和编辑键。其中包括:

<Clear display>	清除屏幕和命令行。如果滚动了屏幕，则仅清除从光标位置到滚动内存结尾的内容。
<Clear line>	清除从光标位置到命令行结尾的内容。
<Delete line>	清除整个命令行。
<Insert line>	转换当前命令行中的任何功能键命令，然后编辑结果。
<Delete char>	删除光标所在位置的字符。
<Insert char>	在插入和覆盖模式之间切换。
<Up/Down arrow>	从历史信息缓冲区中回调上一个/下一个命令。
<Left/Right arrow>	左右移动光标。
<Home up/down>	将光标移到命令行的开头（或结尾）。
<Tab>	如果不存在 <插入行> 键，则执行 <插入行> 功能（请参阅上文）。否则，如果不存在 **--Help--** 功能键，则执行 **--Help--** 功能（同样请参阅上文）。否则，执行常规的制表符功能。
<Backtab>	将光标移到上一个单词的开头。
<Ctrl-L>	刷新屏幕下部的行，并恢复任何必需的终端模式。

可见的功能键命令

如果启用了 **visibles** 配置选项（请参阅下面的 配置），只要需要新命令， **keysh** 就会在功能键标签上显示配置的功能键命令的列表。这是顶级功能键菜单。

如果用户选择其中的一个功能键命令， **keysh** 会将其命令名插入命令行，然后显示一个子菜单，列出该命令的主要参数和（或）选项。

然后，用户可以按从左到右的顺序选择选项功能键和（或）在参数功能键处输入文本。 **keysh** 会自动导航分层功能键菜单，从而在功能键标签上随时向用户显示一组适当的当前选项。

请注意， **keysh** 检测到输入命令分隔符（如管道或分号）时，将自动重新显示顶级功能键菜单，这样用户就可以将功能键用于命令行上的第一个命令以及后继的命令。

不可见的功能键命令

如果启用了 **invisibles** 配置选项（请参阅下面的 配置），并且 **keysh** 识别输入的传统 HP-UX 命令，它将在功能键标签上再次显示一组适当的当前选项，从而为用户提供最后一次使用功能键的机会。与顶级功能键菜单选项相同，用户可以选择忽略功能键，而直接输入传统的 HP-UX 选项。

备用功能键

如果启用了 **backups** 配置选项（请参阅下面的 配置），每当没有其他功能键可显示时（如正在运行命令时），**keysh** 将显示 备用功能键，并对终端功能键进行适当的编程。这样将提供许多用户可能已经习惯的传统静态功能键控制。

传统 HP-UX 命令

如果用户在 **keysh** 显示其顶级功能键菜单时输入传统的 HP-UX 命令， **keysh** 仅显示备用功能键，并允许用户继续操作。

如果 **keysh** 在随后检测到命令分隔符，则将再次重新显示顶级功能键菜单。

功能键命令语法错误

许多功能键命令为用户提供了一组功能键选项，要求用户 必须从其中选择一个（或者至少一个）选项。如果用户未能这样做， **keysh** 会将其当作语法错误，从而显示错误消息，并且在错误得到更正之前将不接受该命令。

同样，许多功能键命令要求用户输入一个或多个功能键参数，这样命令才具有完整的语义。如果用户未能这样做， **keysh** 同样会将其当作语法错误。

功能键命令重定向

用户可以在功能键命令后追加重定向符号（如小于或大于符号后接文件名）。它们是转换的 HP-UX 命令的追加 *verbatim* 。

将 KEYSH 与终端会话管理器一起使用

在终端会话管理器（请参阅 *tsm*(1) ）下运行时， **keysh** 将显示 **tsm** 功能键，而不是备用功能键。如果需要，可以通过设置 **$KEYTSM** 环境变量来覆盖该交互设置（请参阅下面的 “环境变量”这部分内容）。

在 **tsm** 下运行时， **keysh** 还将在状态行中自动显示 **tsm** 窗口编号。

配置

所有 **keysh** 配置功能均通过顶级 **Keysh_config** 功能键命令或 **kc** 内置命令来进行访问。这些功能包括：

- 添加、放置和删除功能键，
- 指定备用功能键，
- 选择全局选项，
- 选择状态行项目，
- 重新启动 keysh，
- 写入配置更改，以及
- 撤消其他配置更改。

每当用户更改 **keysh** 的配置时， **keysh** 会自动更新用户的 **$HOME/.keyshrc** 文件。在后继调用时， **keysh** 会按照先前的配置对其自身进行重新配置。

添加、放置和删除功能键

使用 **kc softkey add** 命令，可以将任意标准功能键（请参阅下面的 标准功能键定义）添加到顶级功能键菜单中。如果需要，可以使用 **with_label** 选项指定备用功能键标签（通常在加密 HP-UX 命令名位置处）。

缺省情况下，添加的功能键将放置在顶级功能键菜单的最后一个 **--More--** 组的末尾。该放置可以使用 **kc softkey add** 命令的 **and_place** 选项或使用 **kc softkey move** 命令来覆盖。

除了标准的功能键之外，还可以使用 **from_user** 或 **from_file** 选项从自定义功能键文件中添加自定义功能键。有关功能键文件格式的说明，请参阅 *softkeys*(4) 。

请注意，每当从特定功能键文件添加功能键时，该文件中的所有剩余功能键就会自动加载，以用作不可见的功能键命令。也可以使用 **kc softkey add invisibles** 命令加载文件中的所有功能键，用作不可见的功能键命令。

使用 **kc softkey delete** 命令，可以删除顶级功能键菜单中的任何功能键。

指定备用功能键

备用功能键通常在用户的 **$HOME/.softkeys** 文件中指定。基本的备用功能键定义行类似于：

> **backup softkey "**<*softkey*>**" literal "**<*string*>**";**

其中 <*softkey*> 是要显示的功能键标签， <*string*> 是用来将终端功能键编程的文本字符串。最多可以指定八个备用功能键。

请注意，必须首先使用 **kc softkey add backups** 命令显式添加备用功能键，然后 **keysh** 才能将其编程。

选择全局选项

使用 **kc option** 命令可配置各个全局选项，其中包括：

backups 启用或禁用备用功能键编程。

help 启用或禁用 **--Help--** 功能键。

invisibles 启用或禁用对不可见功能键命令的识别。

prompts 启用或禁用提示消息自动生成。启用时，只要用户 必须执行某一操作才能完成当前功能键命令， **keysh** 就会显示提示消息。该消息会简述必需的操作。

selectors 启用或禁用键盘选择器的使用。启用时， **keysh** 将在每个功能键标签中显示大写的选择器字符。如果键入未引用的（大写）字符，则像按其相应功能键一样选择功能键。如果通过任何方式引用选择器字符，则将恢复其传统的含义。选择器键应该在所支持的功能键数量不足的终端上使用。

translations 启用或禁用 HP-UX 命令转换的显示。

visibles 启用或禁用对可见功能键命令的显示和识别。

选择状态行项目

使用 **kc status_line** 命令，可以将各个信息项目配置到屏幕底部显示的状态行中，这些信息项目包括：

host_name 主机名。

user_name 用户名。

current_dir 当前目录。

mail_status 基于 **$MAIL** 环境变量（即 **No mail** 、 **You have mail** 或 **You have new mail** ）的邮件状态。

date 日期。

time 一天中的时间。

此外，如果 **$KEYSH** 环境变量已设置，将始终首先显示在状态行中。

重新启动 Keysh

通过 **kc restart** 命令，可以强制 **keysh** 重新读取 **$HOME/.keyshrc** 文件。该命令通常用于将 **keysh** 更新为另一个窗口中的新配置。

通过 **kc restart default** 命令，也可以强制 **keysh** 删除 **$HOME/.keyshrc** 文件并从缺省用户配置中启动。

写入配置更改

通过 **kc write** 命令，可以强制 **keysh** 重写 **$HOME/.keyshrc** 文件。

撤消其他配置更改

使用 **kc undo** 命令，还可以强制 **keysh** 用其初始内容重写 **$HOME/.keyshrc** 文件，从而撤消自调用 **keysh** 以来做出的所有配置更改。

缩放 Keysh 功能

keysh 提供了一组可缩放的功能，用户可对其进行定制，以满足个人的特殊需要。

对于熟悉 HP-UX 命令名（但不一定熟悉命令选项）的用户或者通常首选使 **tsm** 功能键可见的用户，命令 **kc options visibles off** 将防止 **keysh** 在等待命令时显示其顶级功能键菜单；根据具体的情况，它将适当地显示备用功能键或 **tsm** 功能键（这样，通过编辑 **$HOME/.keyshrc** 文件并删除那些添加可见功能键的行，可以大大减少（ **keysh** 的启动时间）。

对于熟悉 HP-UX 命令选项的用户，命令 **kc options invisibles off** 还将防止 **keysh** 显示不可见功能键命令的功能键菜单。

对于不需要备用功能键的用户，命令 **kc options backups off** 将防止 **keysh** 进行备份功能键编程。

请注意，如果 **visibles** 、 **invisibles** 和 **backups** 均被关闭， **keysh** 将根本 不执行任何功能键处理。 **keysh** 会有效地转变成 Korn-Shell，显示状态行并识别光标移动和编辑键。

举例

将 **od** （请参阅 *od*(1) ）功能键添加到顶级功能键菜单的末尾，并将其标记为 **Octal_dump** ：

kc softkey add od with_label Octal_dump

将 *paste*(1) 功能键添加到顶级功能键菜单的开头，并将其标记为 **Paste** ：

kc softkey add paste and_place as_first_softkey

将文件 **˜rpt/.softkeys** 中的自定义 emacs 功能键添加到顶级功能键菜单中的 **ls** （请参阅 *ls*(1) ）功能键之前：

kc softkey add emacs from_user rpt and_place before_softkey ls

添加文件 **~rpt/.softkeys** 中的所有不可见功能键:

kc softkey add invisibles from_user rpt

添加文件 **$HOME/.softkeys** 中的所有备用功能键:

kc softkey add backups

从顶级功能键菜单中删除 **Edit_file** 功能键:

kc softkey delete Edit_file

禁用 **--Help--** 功能键:

kc options help off

将用户名配置到状态行中:

kc status_line user_name on

将所执行的最后一个命令的退出值配置到状态行中:

KEYSH="\${?#0}"

列出当前目录中前十个最大的文件:

**ls long_format | Sort_lines numerically reverse_order **
starting_at_field 5 | head

标准功能键定义

Copy_files 、 **Move_files** 、 **Print_files**
、 **Set_file_attribs** 、 **Switch** 。

adjust 、 **ar** 、 **bdf** 、 **cal** 、 **cancel** 、 **cat** 、 **cd** 、 **cdb** 、 **chatr** 、 **chgrp** 、 **chmod** 、 **chown** 、 **cmp** 、 **col** 、 **comm** 、 **cpio** 、 **cut** 、 **dd** 、 **df** 、 **diff** 、 **dircmp** 、 **disable** 、 **du** 、 **elm** 、 **enable** 、 **exit** 、 **find** 、 **fold** 、 **grep** 、 **head** 、 **jobs** 、 **kill** 、 **lp** 、 **lpstat** 、 **ls** 、 **mailx** 、 **make** 、 **man** 、 **mkdir** 、 **more** 、 **nm** 、 **nroff** 、 **od** 、 **paste** 、 **pg** 、 **pr** 、 **ps** 、 **remsh** 、 **rlogin** 、 **rm** 、 **rmdir** 、 **sdiff** 、 **set** 、 **shar** 、 **sort** 、 **tail** 、 **tar** 、 **tee** 、 **touch** 、 **tr** 、 **umask** 、 **uname** 、 **vi** 、 **wc** 、 **who** 、 **write** 、 **xd** 、 **xdb** 。

环境变量

TERM 指定 *terminfo*(4) 数据库中定义的终端类型。该变量必须是 **keysh** 的调用环境的一部分，否则，它必须在标准 Korn-Shell 启动文件之一中进行设置。

COLUMNS 指定终端屏幕中的列数（如果不同于 *terminfo*(4) 缺省值）。

LINES 指定终端屏幕中的行数（如果不同于 *terminfo*(4) 缺省值）。

PAGER 指定用于显示帮助的首选寻呼。缺省值是 **more** （请参阅 *more*(1) ）。

TZ 指定在状态行上显示时间和日期所使用的时区。缺省值是 **en_US.roman8** 。

KEYBEL 指定由 **keysh** 发送到终端来响铃的字符序列。缺省值是 **^G** 。

KEYENV 指定备用的 **keysh** 配置文件。缺省值是 **$HOME/.keyshrc** 。

KEYESC 指定在将字符视为终端转义序列的一部分时，在各字符之间允许的最大延迟（毫秒）。缺省值为 350 毫秒。

KEYKSH 如果设置，则指定 **keysh** 应尽可能地仿效 Korn-Shell 的行为。不显示功能键或状态行。该模式对于慢速调制解调器线路特别有用。

KEYLOC 如果设置，指定 **keysh** 应该在输入命令时将终端键盘保留在本地模式。它将仿效 Korn-Shell 的行为。

KEYPS1 如果设置，指定 **keysh** 不应重置 **$PS1** 、 **$PS2** 和 **$PS3** 的初始值。请注意， **$PS1** 必须是常量字符串， **keysh** 才能识别它并提供后继的功能键帮助。

KEYSH 指定要在 **keysh** 状态行中包括的任意文本。

KEYSIM 如果设置，指定 **keysh** 应该始终模拟功能键标签，而不使用 HP 终端上的内置标签。

KEYTSM 如果设置，指定 **keysh** 在 **tsm** 运行时 不应该使用 **tsm** 功能键。这种情况下，用户可以使用 **tsm hotkey** 、备用功能键或 **Switch** 功能键命令（请参阅上面的 标准功能键定义）来切换 **tsm** 窗口。

KSH 差异

keysh 是 *ksh*(1) 的扩展，它包括以下例外:

屏幕更新

keysh 优化了显示输出，可利用可用的终端功能。与经常需要刷新命令行大部分内容的 Korn-Shell 不同， **keysh** 可以在适当的屏幕位置插入或删除字符。

这样， **keysh** 在慢速调制解调器线路上要快得多，尤其是在设置 **$KEYKSH** 环境变量时（请参阅上面的 环境变量）。

Emacs 模式编辑

新增的 **<ESC>v** 命令执行 vi 模式 **v** 命令的功能。

初始的 **^N** 命令回调作为上一个命令执行的历史信息行 之后的历史信息行。通过它可以方便地重复历史命令的序列。

不支持 **gmacs** 编辑模式； **emacs** 编辑模式遵循 **^T** 的 GNU emacs (18.54) 定义。

不支持 **^@** 和 <ESC>**n ^K** 命令。

不支持 **M-**<*letter*> 和 **M-]**<*letter*> 别名功能（代替实际的功能键支持）。

Vi 模式编辑

新增的 **o** 命令执行 emacs 模式 **^O** 命令的功能。

初始的 **j** 命令回调作为上一个命令执行的历史信息行 之后的历史信息行。通过它可以方便地重复历史命令的序列。

不支持 **l** 命令。

不支持 @<字母> 别名功能（代替实际的功能键支持）。

u 命令执行 emacs 式嵌套撤消； **u**<空格> 执行传统的 vi 式撤消。

警告

keysh 要求在 **$HOME/.profile** 文件中正确设置 **$TERM** 环境变量。它还要求在非标准尺寸的终端上运行时正确地设置 **$LINES** 和 **$COLUMNS** 。否则，将导致错误消息或混乱的屏幕显示。

keysh 要求按从左到右的顺序选择选项功能键。编辑命令行时，可能会按错误的顺序备份和插入功能键，这将导致命令错误。

keysh 初始化 **$PS1** 、 **$PS2** 和 **$PS3** ，并将其设置为 只读类型 — 不要将其更改，而应使用 **$KEYSH** 来显示附加的状态信息。

keysh 通常会维护 **$HOME/.keyshrc** 文件，而无需用户干预；但是，有时启动错误可能会出现并持续存在。这种情况下，请执行命令 **kc restart default** （删除该文件并恢复到缺省用户配置）或者执行命令 **kc write** （用当前配置重写该文件）。

keysh 假定 HP-UX 命令未进行大量的别名处理；否则，可能会出现意外的命令转换。

在计算命令行参数数量时， **keysh** 将忽略 Korn-Shell 扩展机制的作用，因此有时会低估所指定参数的实际数量。<ESC>***** emacs 模式或 vi 模式编辑命令通常可用于预扩展这些参数。

由于无法进行后继的命令转换， <ESC>**v** emacs 模式编辑命令和 **v** vi 模式编辑命令不能用于编辑（预转换）功能键命令。

如果添加大量的功能键，可能会使 **keysh** 超出 1 MB 的 Korn-Shell 数据大小限制，从而导致破坏性的行为。

keysh 只能对其 *terminfo*(4) 条目定义 **pfkey** 功能的功能键进行编程；同样，它只能使用其 *terminfo*(4) 条目定义 **pln** 功能（同时指定 **lh** 等于 2）的终端上的硬件功能键标签。

$KEYESC 的缺省值选择来在本地和网络环境中提供合理的响应。如果 keysh 将快速键入的 emacs 模式或 vi 模式编辑命令错误解释为终端转义序列，则可能需要将该值减小。

如果在备用功能键的文字键序列中指定 **\n** （换行符），则将在 HP 终端上导致意外的结果；请改用 **\r** （回车）。

当模拟功能键标签时， **keysh** 不显示 **tsm** 功能键。

显示帮助时，有限数量的环境变量和参数将导出到寻呼中。

外部语言环境影响

环境变量

LANG 确定显示功能键和消息所用的语言。

LC_TIME 确定状态行中日期和时间字符串的格式和内容。

国际代码集支持

支持单字节字符代码集。

作者

keysh 由 HP 和 AT&T 联合开发。

文件

/usr/bin/keysh	主可执行文件
/usr/lib/keysh/builtins	**Keysh_config** 功能键定义文件
/usr/lib/keysh/$LANG/softkeys	标准功能键定义文件
/usr/lib/keysh/$LANG/keyshrc	缺省用户配置文件
/usr/lib/nls/$LANG/keysh.cat	消息清单
$HOME/.keyshrc	用户配置文件
$HOME/.softkeys	用户功能键定义文件

另请参阅

ksh(1)、 tsm(1)、 softkeys(4)、 terminfo(4)。

名称

kill - 向进程发送一个信号；终止进程

概要

kill [**-s** *signame*] *pid* ...

kill [**-s** *signum*] *pid* ...

kill -l

过时版本：

kill -*signame pid* ...

kill -*signum pid* ...

说明

kill 命令向由 *pid* 进程标识符指定的每一个进程发送一个信号。缺省信号是 **SIGTERM** ，它通常会终止不捕获信号或忽略信号的进程。

选项

kill 可识别下列选项：

-l (ell) 列出实现支持的 *signame* 的所有值。使用该选项不会发送任何信号。信号的符号名称（不带 **SIG** 前缀）写入标准输出，以空格和换行符分隔。

-s *signame* 发送指定的信号名称。缺省值为 **SIGTERM** ，编号为 **15** 。 *signame* 可以用大写和（或）小写形式指定，可以带或不带 **SIG** 前缀。这些值可以通过使用 **-l** 选项获得。符号名称 **SIGNULL** 表示信号值为零。请参阅下文的“信号名称和编号”。

-s *signum* 发送指定的十进制信号编号。缺省值为 **15** （对于 **SIGTERM** ）。请参阅下文的“信号名称和编号”。

-*signame* （已过时）。等效于 **-s** *signame* 。

-*signum* （已过时）。等效于 **-s** *signum* 。

操作数

pid 是一个进程标识符，可以是以下无符号或负整数之一：

> 0 进程号。

= 0 所有进程（专用系统进程除外），其进程组 ID 等于发送方的进程组 ID 。

=-1 所有进程（专用系统进程除外），条件是用户拥有适当的权限。否则，所有进程（专用系统进程除外）的实际或有效用户 ID 与发送进程的用户 ID 相同。

<-1 所有进程（专用系统进程除外），其进程组 ID 等于 *pid* 的绝对值，其实际或有效用户 ID 与发送进程的用户 ID 相同。

进程号可以通过 **ps** 命令（请参阅 *ps*(1) ）和某些 Shell 中提供的内置 **jobs** 命令找到。

信号名称和编号

下表列出了一些较常用的信号，这些信号在终端上十分有用。要获得完整的列表和详尽的说明，请参阅头文件 **<signal.h>** 和手册条目 *signal*(5) 。

signum	*signame*	名称	说明
0	***SIGNULL***	Null	检查对 *pid* 的访问
1	***SIGHUP***	Hangup	终止；可以捕获
2	***SIGINT***	Interrupt	终止；可以捕获
3	***SIGQUIT***	Quit	使用核心转储终止；可以捕获
9	***SIGKILL***	Kill	强制终止；不能捕获
15	***SIGTERM***	Terminate	终止；可以捕获
24	***SIGSTOP***	Stop	暂停进程；不能捕获
25	***SIGTSTP***	Terminal stop	暂停进程；可以捕获
26	***SIGCONT***	Continue	运行已停止的进程

SIGNULL (**0**) ，是空信号，调用错误检查，但实际上不发送任何信号。它可用于测试 *pid* 的有效性或存在与否。

SIGTERM (**15**) ，是（缺省的）终止信号，可以被接收进程捕获，允许接收方执行顺序关闭操作或完全忽略该信号。对于顺序操作，此为首选。

SIGKILL (**9**) ，是强行终止信号，强制进程立即终止。由于 **SIGKILL** 不能被捕获或忽略，它对终止不响应 **SIGTERM** 的进程很有用。

接收进程必须属于发送进程的用户，除非该用户拥有适当的权限。

作为一种特殊情况，继续信号 SIGCONT 可以被发送到那些是发送进程同一会话成员的任何进程。

返回值

完成时， **kill** 返回以下值之一:

- **0** 对于每一个 *pid* 操作数，至少找到一个匹配进程，并且至少为一个匹配进程成功处理了指定的信号。
- **>0** 发生了错误。

举例

命令:

kill 6135

用信号通知进程号为 6135 的进程终止。这为该进程提供了一个正常退出（删除临时文件等）的机会。

下列等效命令:

kill -s SIGKILL 6135

kill -s KILL 6135

kill -s 9 6135

kill -SIGKILL 6135

kill -KILL 6135

kill -9 6135

通过向进程发送 **SIGKILL** 信号，突然终止进程号为 6135 的进程。这将告知内核立即删除该进程。

警告

如果一个进程在某些操作（例如 I/O）中挂起，因而从未被调度过，则该进程无法终止，直到允许它运行。因此，这样的进程在强行终止后可能永远不会消失。同样，死进程（请参阅 *ps*(1) ）可能已经执行完，但仍保留在系统中，直到它们的父进程捕获它们（请参阅 *wait*(2) ）。使用 **kill** 向它们发送信号无效。

一些非 HP-UX 实现仅将 **kill** 作为 Shell 内置命令提供。

相关内容

本手册条目介绍了 POSIX Shell 的外部命令 **/usr/bin/kill** 和内置命令 **kill** （请参阅 *sh-posix*(1) ）。其他 Shell，例如 C 和 Korn （请分别参阅 *csh*(1) 和 *ksh*(1) ），也将 **kill** 作为一个内置命令提供。这些内置命令的语法和输出可能不同。

另请参阅

csh(1)、 ksh(1)、 ps(1)、 sh(1)、 sh-posix(1)、 kill(2)、 wait(2)、 signal(5)。

符合的标准

kill：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

kinit - 获取并缓存 Kerberos 凭证授予凭证

概要

kinit [**-l** *life_time*] [**-s** *start_time*] [**-v**] [**-p**] [**-f**] [**-k** [**-t** *keytab_filename*]] [**-r** *renewable_life*] [**-R**] [**-c** *cache_filename*] [**-S** *service-name*] [*principal*]

说明

kinit 可获取并缓存 *principal* 的初始凭证授予凭证。

选项

-l *life_time*
请求一个具有由 *life_time* 定义的生命周期的凭证。 *life_time* 的值后必须紧跟下列定界符之一:

s 秒
m 分
h 小时
d 天

例如， **kinit -l 90m** 表示 90 分钟。不能将各种单位混合在一起使用；值 **3h30m** 会导致出错。

如果未指定 **-l** 选项，则使用凭证的缺省生命周期（由各站点配置）。若为凭证指定了比最大凭证生命周期（在各站点中配置）更长的生命周期，则凭证的生命周期将等于最大凭证生命周期。

-s *start_time*
请求事后填写日期的凭证，于 *start_time* 正式启动。 *start_time* 的值后必须紧跟下列定界符之一:

s 秒
m 分
h 小时
d 日

事后填写日期的凭证以 无效标志集发布，且需要在使用前反馈给 Kerberos KDC（密钥分发中心）。

-v
请求将缓存中的凭证授予凭证（具有 无效标志集）传送给 KDC 以进行验证。如果凭证在其所请求的时间范围内，缓存会被通过验证的凭证所替代。

-p
请求可代理凭证。

-f
请求可转发的凭证。

-r *renewable_life*
请求可续订的凭证，其总生命周期为 *renewable_life* 。 *renewable_life* 的值后必须紧跟下列定界符之一:

s 秒
m 分
h 小时
d 天

-R 请求凭证授予凭证的续订。注意，过期的凭证无法续订，即便该凭证仍然处于其可延续生命周期中也是如此。

-k [**-t** *keytab_filename*]
请求主机凭证，从本地主机的 keytab 文件的密钥中获得，可以通过 **-t** *keytab_filename* 选项指定 keytab 文件的名称和位置，否则将使用缺省文件名与位置。

-c *cache_filename* 将 *cache_filename* 用作凭证的缓存名称和位置。如果未使用该选项，则使用缺省缓存名和位置。

缺省凭证缓存可能随系统不同而异。如果设置了 **KRB5CCNAME** 环境变量，则其值可用于为缺省凭证缓存命名。缓存中的任何现有内容都会被 **kinit** 所破坏。

-S *service_name* 指定在获得初始凭证时使用的另一个服务名称。

principal 使用现有缓存中的主体（如果有的话）的名称。

kinit 支持 **[appdefaults]** 部分。在这里指定的关系可为命令行选项所覆盖。 **kinit** 在 **[appdefaults]** 部分中 支持如下关系：

forwardable 该关系用于指定用户是否能够获得可转发的凭证。可设置的有效值为： **true** 、 **false** 、 **yes** 、 **y** 、 **no** 、 **n** 、 **on** 、 **off**。

proxiable 该关系用于指定用户是否能够获得可代理的凭证。可设置的有效值包括： **true** 、 **false** 、 **yes** 、 **y** 、 **no** 、 **n** 、 **on** 、 **off** 。

tkt_lifetime 该关系用于指定可获取的凭证的生命周期。生命周期的单位是秒、分、小时或天。

renew_lifetime 该关系用于指定将获取的凭证的可延续生命周期。生命周期的单位是秒、分、小时或天。

注释

对于 DCE 操作请使用 **/opt/dce/bin/kinit** 。

外部语言环境影响

环境变量

kinit 使用以下环境变量：

KRB5CCNAME 凭证缓存的位置。

文件

/tmp/krb5cc_{*uid*} 缺省凭证缓存。 {*uid*} 是用户的十进制 UID。

/etc/krb5.keytab 本地主机的 keytab 文件的缺省位置。

作者

kinit 由麻省理工学院开发。

另请参阅

kdestroy(1)、 klist(1)、 libkrb5(3)、 kerberos(5)。

名称

klist - 列出缓存的 Kerberos 凭证

概要

klist [**-e**] [[**-c**] [**-f**] [**-s**] [*cache_filename*]] [**-k** [**-t**] [**-K**] [*keytab_filename*]]

说明

klist 可列出凭证缓存中的 Kerberos 主体和 Kerberos 凭证，或 keytab 文件中包含的密钥。

选项

-e 显示凭证缓存中每个凭证的会话密钥及凭证的加密类型，或 keytab 文件中的每个密钥。

-c 列出凭证缓存中包含的凭证。如果 **-c** 或 **-k** 都未指定，则缺省为该选项。

-f 用下面的简略形式显示凭证中存在的标志：

- **F** 可转发
- **f** 已转发
- **P** 可代理
- **p** 代理
- **D** 可事后填写日期
- **d** 事后填写日期的
- **R** 可续借
- **I** 初始
- **i** 无效
- **A** 预先验证的
- **H** 硬件验证的

-s 使 **klist** 以静默状态运行（不生成输出），但是仍然根据是否找到了凭证缓存来设置退出状态。如果 **klist** 找到了凭证缓存，则退出状态为“0”，否则为“1”。

-k 列出 keytab 文件所包含的密钥。

-t 显示 keytab 文件中每个 keytab 条目的输入时间戳。

-K 显示 keytab 文件中 keytab 条目的加密密钥值。

如果不指定 *cache_filename* 或 *keytab_filename* ，则 **klist** 将根据需要显示缺省凭证缓存或 keytab 文件中的凭证。如果设置了 **KRB5CCNAME** 环境变量，则其值将用于命名缺省凭证缓存。

注释

对于 DCE 操作，应使用 **/opt/dce/bin/klist** 。

环境

klist 使用以下环境变量：

KRB5CCNAME
凭证缓存的位置。

文件

/tmp/krb5cc_{*uid*} 缺省的凭证缓存。{*uid*} 为用户的十进制 UID。

/etc/krb5.keytab keytab 文件的缺省位置。

作者

klist 由麻省理工学院开发。

另请参阅

kdestroy(1)、 kinit(1)、 kerberos(5)。

名称

kpasswd - 更改用户的 Kerberos 口令

概要

kpasswd [*principal*]

说明

kpasswd 命令用于更改 Kerberos 主体的口令。 **kpasswd** 提示您输入当前的 Kerberos 口令，该口令可用于从 KDC（密钥分配中心）为用户的 Kerberos 领域获得 **changepw** 凭证。如果 **kpasswd** 成功获得了 **changepw** 凭证，则会提示用户输入两次新口令，从而实现口令更改。

如果主体受某一策略的约束，其中规定了新口令所需的字符类别的长度和（或）数目，那么新口令必须遵循此策略。这五种字符类别分别是小写字母、大写字母、数字、标点符号和所有其他字符。

选项

principal 更改 Kerberos 主体 *principal* 的口令。 **kpasswd** 使用现有缓存中的主体名称（如果有的话）。否则，将从调用 **kpasswd** 命令的用户的标识派生得到主体。

注释

kpasswd 首先在当前领域下的 **krb5.conf** 文件的 [realms] 部分中查找 **kpasswd_server = host:port** 。如果找不到，则 **kpasswd** 会查找 admin_server 项，并用 464 替换端口。

文件

/etc/krb5.conf Kerberos 配置文件。

作者

kpasswd 由麻省理工学院开发。

另请参阅

krb5.conf(4)、 kerberos(5)。

名称

ksh、rksh - Shell，标准（或受限）命令编程语言

概要

ksh [**-aefhikmnoprstuvx**] [**+aefhikmnoprstuvx**] [**-o** *option*]... [**+o** *option*]...
[**-c** *string*] [*arg*]...

rksh [**-aefhikmnoprstuvx**] [**+aefhikmnoprstuvx**] [**-o** *option*]... [**+o** *option*]...
[**-c** *string*] [*arg*]...

说明

ksh 是命令编程语言，执行从终端或文件读取的命令。 **rksh** 是命令解释程序 **ksh** 的受限版本，用于设置登录名和执行环境，其功能相对于标准 Shell 更受控。有关命令行选项和参数的详细信息，尤其是 **set** 命令，请参阅本条目下稍后部分的调用 ksh 和特殊命令。

定义

元字符　下列字符之一:

; & ( ) | < > 换行符　空格　制表符

空白　制表符或空格。

标识符　以字母或下划线开头的字母、数字或下划线序列。标识符用于 函数的名称和 “命名参数”。

单词　由一个或多个没有加引号的元字符分隔的字符序列。

命令　以 Shell 语言语法排列的字符序列。 Shell 通过直接或调用单独实用程序读取命令并执行所需操作。

“特殊命令”　由 Shell 执行的命令，无需创建单独进程。通常称为“内置命令”。除了已经记录的副作用，大部分特殊命令可以作为单独的实用程序实现。

#　**#** 字符是注释的开始。请参阅下面的引用。

命令

简单命令是由空格分隔的单词序列，其前导可以是参数分配列表。（请参阅下文的环境）。第一个单词指定执行命令的名称。除非如下文所示，否则剩余单词作为参数传递给调用的命令。命令名称作为参数 0 传递（请参阅 *exec*(2)）。简单命令的 值正常终止时为其退出状态，在非正常终止时为 200+*status* （8 进制）（有关状态值的列表，请参阅 *signal*(5)）。

管道线是由 **|** 分隔的一个或多个 命令序列。除最后一个命令以外，每个命令的标准输出通过管道（请参阅 *pipe*(2)）与下一命令的标准输入相连接。每一命令作为独立的进程运行； Shell 等待最后命令以终止。管道线的退出状态为管道线的最后命令的退出状态。

列表是由 **;** 、 **&** 、 **&&** 或 **||** 分隔的一个或多个管道线序列，并且可选择由 **;** 、 **&** 、或 **|&** 终止。这五个符号， **;** 、 **&** 和 **|&** 具有相同的优先级。 **&&** 和 **||** 优先级相同，比前三个高。分号 (;) 将使前置管道线按顺序执行；(&) 符号将使前置管道线的异步执行（也就是，Shell 不等待管道线完成）。符号 **|&** 使前置命令或管道线与父 Shell

建立一个双向管道异步执行（称为 联合过程）。衍生命令的标准输出和输入可以使用特殊命令 **read** 和 **print** 的 **-p** 选项向父 Shell 写入或读取，将在后文中介绍。符号 **&&** (**||**) 使前置管道线返回零（非零）值时执行后续的 *list*。在 *list* 中可出现任意数目的换行，取代分号，用于界定命令。

命令可以为简单命令，也可以为下列之一。除非另外说明，命令返回的值为命令中最后执行的简单命令的值。

for *identifier* [**in** *word* ...] **do** *list* **done**

每次执行 **for** 时， *identifier* 设置为从 **in** *word* 列表取得的 *word* 。如果 **in** *word* ... 省略，则 **for** 对每一位置参数设置均执行 **do** *list* （请参阅下文参数替换）。列表中无单词时执行终止。

select *identifier* [**in** *word...*] **do** *list* **done**

select 命令输出标准错误（文件描述符 2）、 *word* ，都带有前置编号。如果 **in** *word* ... 省略，使用位置参数替换（请参阅下文参数替换）。输出 **PS3** 提示符并从标准输入读取一行。如果此行以列表 *word* 编号开始， *identifier* 参数的值根据此编号设置为相应的 *word* 。如果此行为空，重新输出选择列表。否则 *identifier* 参数设置为空。从标准输入中读取的行的内容保存在 **REPLY** 中。对每一选择执行 *list* 直至出现 **break** 或文件结尾标志 (eof)。

case *word* **in** [[( ] *pattern* [| *pattern*] **...**) *list* **;;**] **... esac**

case 命令执行与 *list* 相关联的匹配 *word* 的第一个 *pattern* 。模式的格式与用于文件名称生成的相同（请参阅下文的文件名生成）。

if *list* **then** *list* [**elif** *list* **then** *list*] **...** [**else** *list*] **fi**

执行 **if** 后面的 *list* ，如果返回零退出状态，则执行第一个 **then** 后面的 *list* 。否则执行 **elif** 后面的 *list* ，如果该值为零，执行下一个 **then** 后面的 *list* 。如果失败，执行 **else** *list* 。如果没有执行 **else** *list* 或 **then** *list* ， **if** 返回零退出状态。

while *list* **do** *list* **done**
until *list* **do** *list* **done**

while 命令重复执行 **while** *list* ，如果列表中最后命令的退出状态为零，执行 **do** *list* ；否则循环终止。如果没有执行 **do** *list* 中的命令， **while** 返回零退出状态， **until** 可用于 **while** 以取消循环终止测试。

(*list*) 在单独的环境中执行 *list* 。如果需要两个临近的打开括号用于嵌套，其间需要插入空格用于避免下面说明的算术计算。

{ *list* **;**} 执行 *list* ，但不是在单独环境中。请注意 **{** 是关键字，需要结尾空白用于识别。

[[*expression* **]]** 当 *expression* 成立时，计算 *expression* 并返回零退出状态。有关 *expression* 的说明，请参阅下文的条件表达式。请注意 **[[** 和 **]]** 为关键字，在它们和 *expression* 之间需要空格。

function *identifier* **{** *list* **;}**
identifier **()** **{** *list* **;}**

定义 *identifier* 引用的函数。函数主体为 *list* 中 **{** 和 **}** 之间的命令（请参阅下文的函数）。

time *pipeline* 执行 *pipeline* 并且所用时间、用户时间和系统时间输出为标准错误。请注意 **time** 关键字可以出现在 *pipeline* 的任何位置，用于 *pipeline* 计时。欲在 *pipeline* 中为特定命令计时，请参阅

time(1)。

在不加引号时，下列关键字仅被识别为命令的第一个单词：

if then else elif fi case esac for while

until do done { } function select time [[ ]]

注释

以 **#** 开始的单词将使该单词直至下一行之前的所有后续字符被忽略。

别名

如果已经定义了该命令的别名，则 别名文本会替换命令的第一个单词。“别名”可以由除元字符之外的所有字符组成，引用字符、文件扩展字符、参数和命令替换字符，及 **=** 。替代字符串可以包含所有有效 Shell 脚本，包括上面列举的元字符。除了正在替代的部分，每个命令替代文本的第一个单词作为附加别名进行测试。如果别名的最后一个字符值为 空白，别名后的单词也作为别名替换检查。别名可用于重新定义专用的内置命令，但不能用于重新定义上面列举的关键字。使用 **alias** 命令可以创建、列举和导出别名，使用 **unalias** 命令可以删除别名。导出别名在子 Shell 中功能保留，但对于 Shell 的单独调用必须重新初始化（请参阅下文的调用 ksh ）。

别名在读取脚本时执行，而不是在脚本执行时。因此，要使其生效， **alias** 必须在读取引用别名的命令之前执行。

别名经常用作完整路径名的快捷方式。别名功能的一个选项是允许别名值自动设置为相应命令的完整路径名。该别名称为 “跟踪别名”。跟踪别名的值在第一次读取标识符时定义，在重新设置 **PATH** 变量时成为未定义。该别名仍被跟踪，因而下一次引用时重定义值。多个跟踪别名编译生成 Shell。 **set** 命令的 **-h** 选项将名称为 *identifier* 的命令转换为跟踪别名。

下列 “输出别名”编译生成 Shell，但是可能未设置或重定义：

autoload='typeset -fu'
false='let 0'
functions='typeset -f'
hash='alias -t -'
history='fc -l'
integer='typeset -i'
nohup='nohup '
r='fc -e -'
stop='kill -STOP'
suspend='kill -STOP $$'
true=':'
type='whence -v'

波浪符替换

执行波浪符替换后，检查每个单词是否以未用引号引起来的 ˜ 开头。如果是，检查 **/** 之前的单词是否与 **/etc/passwd** 文件中的用户名匹配。如果发现匹配，匹配用户的登录目录替代 ˜ 和匹配的登录名。这称为波浪符替

换。如果未发现匹配，保留原文不变。单独使用或在 **/** 之前的 ˜ 用 **HOME** 参数的值替代。后面紧跟 **+** 或 **-** 的 ˜ 分别用 **PWD** 和 **OLDPWD** 参数的值替代。此外，当参数分配以 ˜ 开头时尝试本地交换。

命令替换

以美元符号 ($(命令)) 或两个反单引号（着重号）（‘命令‘ ）开头的括号内命令的标准输出可用作部分或整个单词；结尾换行符删除。在第二种格式（已停止使用）中，引号中间的字符串在命令执行之前作为特殊引用字符处理（请参阅下文引用）。命令替换 **$(cat file)** 可用 **$(<file)** 取代，速度更快。大多数不执行 I/O 重定向的特殊命令（内置）的命令替换不创建单独进程即执行。但是，函数的命令替换创建单独进程以执行函数和该函数中所有命令（内置或其他）。

以美元符号开头的双括号内的算术表达式（ $((表达式)) ）在双括号内用算术表达式的值替换（有关算术表达式的说明，请参阅下文的算术计算）。

参数替换

参数可以为 标识符、一位或多位数字、或任意字符 ***** 、 **@** 、 **#** 、 **?** 、 **-** 、 **$** 和 **!** 。“命名参数”（由标识符表示的参数）有一个值和零或多属性。命名参数可以使用 **typeset** 特殊命令指定值和属性。支持 **ksh** 的属性将在下文的 **typeset** 特殊命令说明。将参数传递值和属性导出到环境。

Shell 支持有限的一维数组设备。数组参数的元素通过下标引用。下标由 **[** 之后的算术表达式表示（请参阅下文算术计算），后面紧跟 **]** 。要为数组分配值，请使用 **set -A** *name value ...* 。所有下标的值必须在 **0** 到 **1023** 之间。数组不需要声明。任何引用具有有效下标的命名参数均合法并在需要时创建一个数组。不使用下标引用数组等效于引用第一个元素。

命名参数的值也可以通过写入分配：

*name***=***value* [*name***=***value*]...

如果为 *name* 设置了 **-i** 整数类型， *value* 受算术值限制，如下文所述。

位置参数以数字表示，使用 **set** 特殊命令分配值。参数 **$0** 在调用 Shell 时用于从参数零设置。

字符 **$** 用于引入可替换的 *parameters* 。

${***parameter***}** 如果有的话，替换参数值。当字母、数字或下划线在 *parameter* 之后且未解释为名称的一部分时、或者当命名参数为下标时，需要添加括号。如果 *parameter* 为一位或多位数字，则其为位置参数。超过一位数字的位置参数必须在括号内。如果 *parameter* 为 ***** 或 **@** ，则替换所有以 **$1 开头的位置参数（用字段分隔符字符分隔）。如果数组的 *identifier* 下标使用了 ***** 或 **@** ，则替换所有元素的值（用字段分隔符分隔）。 Shell 读取从 **${** 开始到匹配的 **}** 之间的全部字符，将其作为相同单词的一部分，即使其中包含括号或元字符。

${#*parameter***}** 如果 *parameter* 为 ***** 或 **@** ，则替换位置参数的数字。否则，替换 *parameter* 值的长度。

${#*identifier***[*]}** 替换数组元素的数字 *identifier* 。

${_parameter_**:-**_word_**}** 如果 _parameter_ 设置为非空则替换其值，否则替换 _word_ 。

${_parameter_**:=**_word_**}** 如果 _parameter_ 未设置或为空，将其设置为 _word_ ，然后替换参数值。位置参数无法采用此方法分配。

${_parameter_**:?**_word_**}** 如果 _parameter_ 设置为非空则替换该值，否则输出 _word_ 并退出 Shell。如果 _word_ 省略，则输出标准信息。

${_parameter_**:+**_word_**}** 如果 _parameter_ 设置为非空则替换 _word_ ，否则不替换。

${_parameter_**#**_pattern_**}**

${_parameter_**##**_pattern_**}**

如果 Shell _pattern_ 与 _parameter_ 开始值匹配，此替换值为 _parameter_ 值，并删除匹配部分，否则替换 _parameter_ 值。前者删除最少匹配模式；后者删除最大匹配模式。

${_parameter_**%**_pattern_**}**

${_parameter_**%%**_pattern_**}**

如果 Shell _pattern_ 与 _parameter_ 开始值匹配，此替换的值为 _parameter_ 值，并删除匹配部分，否则替换 _parameter_ 值。前者删除最少匹配模式；后者删除最大匹配模式。

上文中不计算 _word_ ，除非将其作为替换字符串。所以在下例中， **pwd** 仅在 **d** 未设置或为零时执行：

echo ${d:-$(pwd)}

如果上述表达式中省略冒号 (:)， Shell 仅检查以确定是否设置了 _parameter_ 。

下列参数由 Shell 自动设置：

位置参数的数字是十进制的。

- 在调用时或使用 **set** 命令时， Shell 提供选项。

? 最后执行的命令返回十进制值。

$ 此 Shell 的进程号。

_ 最初， _ 的值为 _environment_ 中传递的 Shell 或执行脚本的绝对路径名。接下来设置前一条命令的最后一个参数。此参数非异步命令设置。此参数用于检查邮件时存放匹配 **MAIL** 文件的名称。

! 调用最后的背景命令的进程号。

COLUMNS 如果设置了此变量，其值用于定义 Shell 编辑模式下编辑窗口的宽度和输出 **select** 列表。在窗口环境中，如果 Shell 检查到窗口大小更改， Shell 更新 **COLUMNS** 值。

ERRNO 最近失败的系统调用设置 **errno** 的值。此值与系统相关，用于调试目的。

LINENO 执行脚本或函数中当前行的行号。

LINES 如果设置了此变量，该值用于确定 **select** 列表输出列长度。 **select** 列表竖直输出直至大约三分之二 **LINES** 行被填充。在窗口环境中，如果 Shell 检测到窗口大小更改，则 Shell 会更新 **LINES** 值。

OLDPWD **cd** 命令设置前一工作目录。

OPTARG **getopts** 特殊命令处理最后选项参数值。

OPTIND **getopts** 特殊命令处理最后选项参数索引。

PPID 父 Shell 的进程号。

PWD **cd** 命令设置当前工作目录。

RANDOM 每次计算该参数时生成一个随机整数，分配为 0 到 32767 之间的数字。随机数字的序列可以通过为 **RANDOM** 参数分配一个数值初始化。

REPLY 不提供参数时，使用 **select** 语句和 **read** 特殊命令设置此参数。

SECONDS 每次引用此参数时，返回从调用 Shell 开始的秒数。如果此参数分配了值，引用返回值为分配值加上从分配开始的秒数。

Shell 使用下列参数:

CDPATH **cd** 命令的搜索路径。

EDITOR 如果此变量的值末尾为 **emacs** 、 **gmacs** 或 **vi** ，并且未设置 **VISUAL** 变量，则打开相应选项（请参阅下文特殊命令中的 **set** ）。

ENV 如果设置了此参数，当 Shell 调用时（请参阅下文的调用 ksh ），参数替换会按值执行以生成欲执行脚本的路径名。此文件通常用于 *alias* 和 *function* 定义。

FCEDIT 用于 **fc** 命令的缺省编辑器名称。

FPATH 用于函数定义的搜索路径。引用的函数具有 **-u** 属性以及找不到命令的情况下搜索此路径。如果找到可执行文件，则在当前环境下读取并执行。

IFS 内部字段分隔符，通常为 *space* 、 *tab* 和 *newline* ，用于分隔命令或参数替换产生的命令单词，以及用于分隔特殊命令 **read** 的单词。 **IFS** 参数的第一个字符用于分隔 **"$*"** 替换的参数（请参阅下文的引用）。

HISTFILE 如果此参数在 Shell 调用时设置，该值为用于存储命令历史记录的文件的路径名。缺省值为 **$HOME/.sh_history** 。 如果用户具有适当权限并且未提供 **HISTFILE** ，则不使用历史记录文件（请参阅下文的命令再输入）。

HISTSIZE 如果此参数在 Shell 调用时设置，则 Shell 可访问的先前输入命令的数字将大于等于此数字。缺省值为 **128** 。

HOME **cd** 命令的缺省参数（home 目录）。

MAIL 如果此参数设置为邮件文件的名称，并且 **MAILPATH** 参数未设置，则 Shell 在指定文件中通知用户已接收到邮件。

MAILCHECK 此变量指定（以秒为单位） Shell 检查任意文件发生更改的修改时间的频率，该文件由 **MAILPATH** 或 **MAIL** 参数指定。缺省值为 **600** 秒。经过这段时间后， Shell 在发出下一提示前进行检查。

MAILPATH 由冒号分隔的 (:) 文件名列表。如果设置了此参数， Shell 通知用户指定文件的修改，该修改发生在过去的 **MAILCHECK** 秒内。每个文件名其后均可为 **?** 和可打印消息，这时消息使用参数 **$_** 进行参数替换，定义为更改文件的名称。缺省值为 **you have mail in $_**。

PATH 命令的搜索路径（请参阅下文执行）。用户执行 **rksh** 时无法改变 **PATH** （除了在 **.profile** 文件中）。

PS1 此参数扩展用于参数替换，定义主提示字符，缺省值为 **$** 其后跟一个空格字符。在主提示字符串中的字符 **!** 由命令数字替换（请参阅下文的命令再输入）。欲在提示符中包含 **!** ，请使用 **!!** 。

PS2 二级提示字符串，缺省值为 **>** ，其后跟一个空格字符。

PS3 在 **select** 循环中使用选择提示字符串，缺省值为 **#?** ，其后跟一个空格字符。

PS4 此变量的值扩展用于参数替换和执行跟踪时置于每一行的开头。如果未设置 **PS4** ，则执行跟踪提示符为 **+** ，其后跟一个空格字符。

SHELL Shell 的路径名称保存在环境中。调用 Shell 时，如果此变量值的基名中有 **r** ，则该 Shell 受限。

TMOUT 如果设置值大于零，在发出 **PS1** 提示后的指定秒数内未输入命令，则 Shell 终止。

VISUAL 此变量的值以 *emacs* 、 *gmacs* 或 *vi* 结束时调用相应选项（请参阅下文特殊命令中的 **set**）。

Shell 为 **PATH**、 **PS1**、 **PS2**、 **MAILCHECK**、 **TMOUT**, 提供缺省值，但 Shell 不会自动设置 **IFS**、 **HOME**、 **SHELL**、 **ENV** 和 **MAIL** （虽然 **HOME**、 **SHELL** 和 **MAIL** 被 *login*(1) 设置）。

空白解释

在参数和命令替换之后，扫描替换结果中的字段分隔符字符（在 **IFS** 中查找），并在找到该字符时拆分为明确的参数）。 **ksh** 保留明确为空的参数（ 或 **''** ）但删除隐含空的参数（由 *parameters* 生成，没有值）。

文件名生成

替换之后，命令 *word* 以文件名扩展模式处理，除非 **-f** 选项为 **set** 。模式的格式为 *regexp*(5) 定义的模式匹配表示法。使用匹配该模式的排序文件名替换单词。如果没有发现匹配该模式的文件名，则保留该单词。

regexp(5) 所说明的表示法之外， **ksh** 识别由一个或多个模式列表组成的复合模式，模式之间用 **|** 分隔。复合模式

可由下列一个或多个组成:

?(*pattern-list***)** 可选匹配任何给定模式。

(pattern-list***)** 匹配零次或多次出现的给定模式。

+(*pattern-list***)** 匹配一次或多次出现的给定模式。

@(*pattern-list***)** 精确匹配一个给定模式。

!(*pattern-list***)** 匹配除给定模式之外的全部模式。

引用

上面列举的 *metacharacters* （请参阅上文定义）对 Shell 具有特殊含义，并导致单词的终止，除非加引号。通过在字符前加上 **** ，字符可以被 “加引号” （也就是代表自身）。忽略 *****newline* 对。所有在一对单引号 (**' '**) 中的字符被引用。在多个单引号内不能出现单个单引号。双引号内 （**"..."**） 发生参数和命令替换， **** 引用字符 **** 、 **‘** 、 **"** 和 **$** 。 **$*** 和 **$@** 在不被引用、用作参数分配值或文件名称时具有相同的含义。但使用命令参数时， **"$*"** 与 **"$1***d***$2***d***..."** 相同，此处 *d* 为 **IFS** 参数第一个字符，而 **"$@"** 与 **"$1" "$2"** 相同。在反单引号（着重号）(**‘..‘**) 中的标记， **** 引用字符 **** 、 **‘** 和 **$** 。如果反单引号出现在双引号中， **** 仍旧引用字符 **"**。

关键字和别名的特殊含义可以通过引用关键字的任意字符删除。下面列举的函数名或特殊命令的识别不能通过引用更改。

算术计算

特殊命令 **let** 提供整数运算功能。使用长运算进行计算。常量使用 [*base***#]***n* 的格式，此处 *base* 是代表运算基数的 2 到 63 之间的十进制数，以 *n* 为基数的数字。如果 *base* 省略，基数为 10。

算术表达式使用与 C 语言相同的语法、优先级和组合规则。支持除 **++** 、 **--** 、 **?:** 和 **,** 之外的所有整数运算符。可以在算术表达式中使用名称引用变量，而不需要参数替换语法。引用变量时，其值作为算术表达式计算。

variable 的内部整数表示可以使用 **typeset** 特殊命令的 **-i** 选项指定。算术运算根据具有 **-i** 属性的变量所分配的值执行操作。如果用户未指定运算基数，分配给第一个变量的值即为运算基数。发生参数替换时使用基数。

由于多种运算符需要引用，提供了 **let** 命令的替换格式。任何以 **((** 开始的命令，一直到匹配的 **))** 之间的字符当作引用表达式处理。具体的说， **((**...**))** 与 **let "** ...**"**.

提示符

使用交互方式时，Shell 读取命令之前提示 **PS1** 的值。任何时候如果输入新行并需要输入更多以完成命令时，二级提示符 (**PS2** 的值）显示。

条件表达式。

“条件表达式”与 **[[** 复合命令共同使用，用于测试文件属性和比较字符串。不执行 **[[** 和 **]]** 之间单词的单词分割和文件名生成。每个表达式可以使用一个或多个下列一元或二元表达式构建:

如果 *file* 存在时 **-a** *file* 为真。

-b *file* 如果 *file* 存在且为块设备专用文件时为真。

-c *file*	如果 *file* 存在且为字符设备专用文件时为真。
-d *file*	如果 *file* 存在且为目录时为真。
-f *file*	如果 *file* 存在且为普通文件时为真。
-g *file*	如果 *file* 存在且设置了 setgid 位时为真。
-h *file*	如果 *file* 存在且为符号链接时为真。
-k *file*	如果 *file* 存在且设置了粘着位时为真。
-n *string*	如果 *string* 长度不为零时为真。
-o *option*	如果使用了选项命名 *option* 时为真。
-p *file*	如果 *file* 存在且为 fifo 专用文件或管道时为真。
-r *file*	如果 *file* 存在且当前进程可读取时为真。
-s *file*	如果 *file* 存在且大小超过零时为真。
-t *fildes*	如果文件描述符数字 *fildes* 打开且与终端设备关联时为真。
-u *file*	如果 *file* 存在且设置了 setuid 位时为真。
-w *file*	如果 *file* 存在且当前进程可写入时为真。
-x *file*	如果 *file* 存在且当前进程可执行时为真。如果 *file* 存在且为目录、并允许当前进程搜索目录时为真。
-z *string*	如果 *string* 长度为零时为真。
-L *file*	如果 *file* 存在且为符号链接时为真。
-O *file*	如果 *file* 存在且为此进程有效用户 ID 所有时为真。
-G *file*	如果 *file* 存在且组可以与此进程有效组 ID 匹配时为真。
-S *file*	如果 *file* 存在且为套接字时为真。
file1 **-nt** *file2*	如果 *file1* 存在且比 *file2* 新时为真。
file1 **-ot** *file2*	如果 *file1* 存在且比 *file2* 旧时为真。
file1 **-ef** *file2*	如果 *file1* 和 *file2* 存在并引用相同文件时为真。
string **=** *pattern*	如果 *string* 匹配 *pattern* 时为真。
string **!=** *pattern*	如果 *string* 不匹配 *pattern* 时为真。
string1 **<** *string2*	如果 *string1* 的字符 ASCII 值在 *string2* 之前时为真。
string1 **>** *string2*	如果 *string1* 字符 ASCII 值在 *string2* 之后时为真。
exp1 **-eq** *exp2*	如果 *exp1* 等于 *exp2* 时为真。
exp1 **-ne** *exp2*	如果 *exp1* 不等于 *exp2* 时为真。
exp1 **-lt** *exp2*	如果 *exp1* 小于 *exp2* 时为真。
exp1 **-gt** *exp2*	如果 *exp1* 大于 *exp2* 时为真。
exp1 **-le** *exp2*	如果 *exp1* 小于或等于 *exp2* 时为真。
exp1 **-ge** *exp2*	如果 *exp1* 大于或等于 *exp2* 时为真。

复合表达式可由这些基元构建如下，按优先级的降序排列。

如果	*expression* 为真时 (*expression*) 为真。用于组表达式。
! *expression*	如果 *expression* 为假时为真。

expression1 **&&** *expression2*	如果 *expression1* 和 *expression2* 均为真时为真。
expression1 **||** *expression2*	如果 *expression1* 或 *expression2* 为真则为真。

输入/输出

命令执行之前，其输入和输出可以使用专用表示法重定向由 Shell 解释。下列内容可以出现在简单命令内的任何位置，或在命令之前之后，并且不传递给调用命令。命令和参数替换发生在使用 *word* 或 *digit* 之前，除非如下所示。文件名生成仅出现在模式匹配单一文件且不执行空白解释时。

<*word* 使用文件 *word* 作为标准输入（文件描述符 **0** ）。

>*word* 使用文件 *word* 作为标准输出（文件描述符 **1** ）。如果不存在该文件，将另行创建。如果文件存在，并且使用 **noclobber** 选项，则发生错误；否则文件被截断为零长度。

>|*word* 与 **>** 相同，区别在于覆盖 **noclobber** 选项。

>>*word* 使用文件 *word* 作为标准输出。如果文件存在，追加输出到其中（通过首先搜索文件末尾）；否则，另行创建文件。

<>*word* 打开文件 *word* 作为标准输入以进行读取和写入。如果不存在该文件，将另行创建。

<<[**-**]*word* 读取 Shell 输入直至出现行与 *word* 匹配，或者抵达文件末尾。在 *word* 上没有执行参数替换、命令替换或文件名生成。得到的文档称为 本文档，作为标准输入。如果引用 *word* 的任意字符，不对文档字符进行解释。否则，发生参数和命令替换，忽略 *****newline*，必须使用 **** 引用字符 **** 、 **$** 、 **‘** 和 *word* 的第一个字符。如果 **-** 追加到 **<<** ，则从 *word* 和文档去掉所有前导制表符。

<&*digit* 从文件描述符复制标准输入 *digit* （请参阅 *dup*(2) ）。

>&*digit* 标准输出复制到文件描述符 *digit* （请参阅 *dup*(2) ）。

<&- 标准输入关闭。

>&- 标准输出关闭。

<&p 来自联合进程的输入移动到标准输入。

>&p 到联合进程去的输出移动到标准输出。

如果上述一项有数字前导，文件描述符号引用由该数字指定（取代缺省的 **0** 或 **1** ）。例如:

... **2>&1**

意味着文件描述符 2 打开，作为文件描述符 1 的副本用于写入。

重定向顺序很重要，因为 Shell 根据当前打开文件在计算时与指定文件描述符的关联计算重定向引用文件描述符。例如:

... **1>***fname* **2>&1**

首先分配文件描述符 1（标准输出）给文件 *fname* ，然后分配文件描述符 2（标准错误）给分配给文件描述符 1

的文件，也就是 *fname* 。另一方面，如果重定向顺序反转如下：

... **2>&1 1>***fname*

文件描述符 2 分配给当前标准输出，（用户终端，除非继承了不同的分配）。此时文件描述符 1 重新分配给文件 *fname* ，不更改文件描述符 2 的分配。

co-process 的输入和输出可移动到多个文件描述符，允许其他命令使用上述重定向运算符向其中写入和读取。如果当前 *co-process* 输入移动到多个文件描述符，另一个 *co-process* 开始。

如果命令后跟随 **&** 并且作业控制非活动状态，命令的缺省标准输入为空文件 **/dev/null** 。否则，执行命令的环境包括调用 Shell 的文件描述符，通过输入/输出规格修改。

环境

环境（请参阅 *environ*(5) ）为传递给执行程序的名称值对列表，类似于普通参数列表。名称必须为 标识符并且值为字符串。 Shell 与环境以多种方式交互操作。调用时，Shell 扫描环境并为找到的每个名称创建参数，赋予相应的值并且标记为 *export* 。执行过的命令继承环境。如果用户修改该参数的值或使用 **export** 或 **typeset -x** 命令创建新参数，该值也成为环境的一部分。执行完的命令看到的环境由从 Shell 继承的所有名称值对组成，其值可以被当前 Shell 修改，此外任何附加必须在 **export** 或 **typeset -x** 命令中说明。

任何 *simple-command* 或函数的环境可以通过添加一个或多个参数分配前缀进行扩充。参数分配扩充采用 *identi-fier***=***value* 格式。例如，

TERM=450 *cmd args*

和

(export TERM; TERM=450; *cmd args***)**

相同（只要采用上述 *cmd* 计算，除非使用下面列举的有百分符号的特殊命令）。

如果设置了 **-k** 选项，在环境中存在全部参数分配，即使是在命令名称之后发生。下列回显语句输出 **a=b c** 。在设置 **-k** 选项之后，第二回显语句仅输出 **c** ：

echo a=b c
set -k
echo a=b c

此功能用于使用脚本写入 Shell 早期版本，在新脚本中强烈建议不要使用。此功能可能某天会 去除。

函数

function 关键字（在上面章节的命令中说明）用于定义 Shell 函数。 Shell 函数在内部读取和存储。函数读取时解析别名。函数的执行与命令相似，参数传递为位置参数（请参阅下文的执行）。

函数在与调用者同一进程中执行，只是函数的命令替换创建一个新进程。函数共享所有文件并提供给调用者工作目录。调用者捕获的陷阱重置为它们在函数内部的缺省操作。如果函数未捕获或明确忽略陷阱条件，函数终止并将状态传递给调用者。在调用者环境下，函数内部设置的 **EXIT** 陷阱在函数运行完毕后执行。正常情况下，变量在调用程序和函数间共享。但是， **typeset** 特殊命令使用函数定义本地变量，其范围包括当前函数和所有调用函

数。

特殊命令 **return** 用于返回函数调用。函数内部的错误将控制权返回给调用者。

函数标识符可以使用 **typeset** 特殊命令的 **+f** 选项列表。功能标识符和功能关联文本可以使用 **-f** 选项列表。使用 **unset** 特殊命令的 **-f** 选项可以取消定义。

正常情况下，当 Shell 执行 Shell 脚本时函数未设置。 **typeset** 命令的 **-xf** 选项允许函数导出到脚本，不需要重新调用 Shell 即可执行。函数必须定义为跨越单独 Shell 调用，并置于 **ENV** 文件中。

作业

如果使用了 **set** 命令的 **monitor** 选项，交互式 Shell 与每个管道线关联 “作业”。保留当前作业表，通过 **jobs** 命令输出，并为之分配小整数值。当作业以 **&** 异步开始时，Shell 输出一行类似于：

[1] 1234

表明作业编号 1 已异步启动并具有一个（最高级）进程，其进程 ID 为 1234。

如果用户返回一个作业并希望完成其他功能，输入挂起字符（通常为 **^Z** (Ctrl-Z)）以发送 STOP 信号至当前作业。 Shell 指示作业为“Stopped”并输出其他提示符。此作业状态可使用 **bg** 命令处理，将其放入后台，运行其他命令（该作业在后台停止或运行的同时），并最终使用 **fg** 命令将作业重新启动或返回至前台。 **^Z** 立即发生作用，与中断类似，这是因为未决的输出和未读取的输入在键入 **^Z** 时被立即忽略。

在后台运行的作业如果试图读取终端则停止。后台作业通常允许生成输出，但可通过 **stty tostop** 命令禁用。如果用户设置该 tty 选项，则后台作业在试图生成输入时停止。

有多种方法引用 Shell 中的作业。在所有作业进程中可以使用进程 ID 调用作业，或下列方式之一：

%*number*	具有给定编号的作业。
%*string*	任何以 *string* 为命令行的开头的作业。
%?*string*	任何命令行中包括 *string* 的作业。
%%	当前作业。
%+	等效于 **%%** 。
%-	先前作业。

Shell 在进程状态更改时立即获知。 Shell 通知用户作业堵塞并被阻止进一步运行，但仅在输出提示之前。

当监视模式打开时，后台中已完成触发陷阱的作业设置 **CHLD** 。

在作业尚在运行或已停止时试图离开 Shell 将出现警告： **You have stopped (running) jobs** 。使用 **jobs** 命令标识它们。立即尝试再次退出将终止停止的作业， Shell 不再次发出警告。

信号

用于调用命令的 INT 和 QUIT 信号在命令跟随 **&** 和关闭 **monitor** 选项时被忽略。否则，信号继承其父 Shell 的值，信号 11 例外（另请参阅下文的 **trap** 命令）。

执行

每次执行命令时都会作替换。如果命令名称与下列特殊命令其中之一匹配，则在当前 Shell 进程中执行。下一步， **ksh** 检查命令名称用于确定是否与某个用户定义的函数匹配。如果是，则 **ksh** 保存位置参数并将其设置为 *function* 调用的参数。 **0** 位置参数设置为函数名称。当 *function* 完成或发送 **return** 时， **ksh** 存储位置参数列表并执行设置在函数内 **EXIT** 上的任何陷阱。 *function* 的值为最后执行命令的值。在当前 Shell 进程中执行函数。如果命令名称不是 “特殊命令”或用户定义 函数， **ksh** 创建一个进程并试图使用 **exec** 执行命令（请参阅 *exec*(2)）。

Shell 参数 **PATH** 定义包含命令的目录的搜索路径。替换目录名称使用冒号分隔 (:)。缺省路径为 **/usr/bin:** （以该顺序指定 **/usr/bin** 和当前目录）。请注意当前目录用空路径名指定，立即出现在等号后和冒号分隔符之间，或者在路径列表的末端。如果文件名包括 **/** ，则不使用搜索路径。否则在路径下的每个目录中搜索可执行文件。如果文件具有执行权限但不是目录或可执行对象代码文件，假定其为脚本文件，即解释程序的数据文件。如果脚本文件的头两个字符为 **#!** 和 **exec** （请参阅 *exec*(2) ），希望后面跟随解释程序的路径名称。然后 **exec** 试图以单独进程执行指定的解释程序，读取全部脚本文件。如果调用 **exec** 失败，衍生的 **/usr/bin/ksh** 用于解释脚本文件。在这种情况下，删除所有非导出别名、函数和命名参数。如果 Shell 命令文件无读取权限，或者如果 *setuid* 和（或） *setgid* 为在文件中设置，Shell 执行代理以设置权限和执行 Shell ，Shell 命令文件作为打开文件传递下去。括号命令也在子 Shell 内执行，不删除非导出量。

命令再输入

文本最后的 **HISTSIZE** （缺省值 128）命令从终端设备输入，保存在 “历史记录”文件中。文件 **$HOME/.sh_history** 在 **HISTFILE** 变量未设置或不可写入时使用。 Shell 可以访问所有使用相同 **HISTFILE** 命名的 “互动式” Shell 命令。特殊命令 **fc** 用于列出或编辑此部分文件。要编辑或列出部分文件可通过编号及给定第一个字符或命令的字符进行选择。可以指定一个或一定范围的命令。如果没有为参数 **fc** 指定编辑器程序，则使用 **FCEDIT** 参数的值。如果未定义 **FCEDIT** ，则使用 **/usr/bin/ed** 。离开编辑器时编辑命令输出并重执行。编辑器名称 **-** 用于跳过编辑阶段和重新执行命令。这种情况下，使用格式为 *old*=*new* 的替换参数在执行之前修改命令。例如，当 **r** 被转换为别名 **fc -e -** ，输入 **r bad=good c** 重新执行最近的以字母 **c** 开始的命令并使用 **good** 字符串更换首先出现的 **bad** 字符串。

历史记录文件在发生所有下列条件时调整：

文件大小超过四千字节。

其中命令数超过 **HISTSIZE** 。

文件在过去的十分钟内没有更改。

用户有历史记录文件驻留目录的写入权限。

如果上述条件不是全部发生，历史记录文件不会调整。历史记录文件调整时，最近的 **HISTSIZE** 命令保存在历史记录文件中。

特殊命令

下列简单命令在 Shell 进程内执行。允许输入/输出重定向。除非另作说明，文件描述符 1 为缺省输出位置和退出状态，如果没有语法错误则为零。前面有 **%** 或 **%%** 的命令采用如下所示特殊方法处理：

1. 命令完成时，命令前置的变量分配列表仍然有效。
2. I/O 重定向在变量分配之后处理。
3. 包含有特定错误的脚本将异常中止。
4. 变量分配格式单词，其后跟随命令、前置 %%，采用与变量分配相同的规则扩展。这意味着本地交换在 **=** 号之后执行，不执行单词分割和文件名生成。

% : [*arg* ...] 此命令仅用于扩展参数。返回零退出代码。

% . *file* [*arg* ...] 从 *file* 读取和执行命令并返回。命令在当前 Shell 环境中执行。 **PATH** 指定的搜索路径用于查找包含 *file* 的目录。如果参数 *arg* 已经给定，则成为位置参数。否则位置参数保持不变。退出状态为最后执行命令的退出状态。不需要为 *file* 设置执行权限位。

%% alias [**-tx**] [*name*[**=***value*] ...]

无参数的 **alias** 输出别名列表，在标准输出上以 *name***=***value* 格式输出。为每个 *value* 给定的 *alias* 定义名称。 *value* 中的结尾空格使检查用于别名替换的下一单词。 **-t** 选项用于设置和列出跟踪别名。跟踪别名的值为相应的给定 *name* 的完整路径名。跟踪别名的值在 **PATH** 值重置的时候成为未定义，但是仍保留对别名的跟踪。不使用 **-t** 选项，对于每个参数列表的 *name* ，如果其 *value* 没有给定，则输出别名的名称和值。 **-x** 选项用于设置或输出导出别名。导出别名的定义跨越子 Shell 环境。别名返回真值，除非没有定义别名而给定了 *name* 。

bg [*job* ...] 将指定的 *jobs* 放入后台。当前作业在 *job* 未指定时放入后台。有关 *job* 格式的说明，请参阅作业。

% break [*n*] 如果存在，从 **for** 、 **while** 、 **until** 或 **select** 封闭循环中退出。如果指定了 *n* ，中断 *n* 级。

% continue [*n*] 继续下一 **for** 、 **while** 、 **until** 或 **select** 封闭循环的迭代。如果未指定 *n* ，继续 *n*-th 封闭循环。

cd [**-L**|**-P**] [*arg*]

cd *old new* 此命令可以采取两种格式之一。在第一种格式中，更改当前目录为 *arg* 。如果 *arg* 为 **-** ，则更改目录为先前目录。 **-L** 选项（缺省值）在处理符号链接时保留逻辑名称。 **cd -L ..** 移动当前目录到离根目录更近的一级。 **-P** 选项在处理符号链接时保留物理路径。 **cd -P ..** 将当前工作目录更改为当前目录的父目录。 Shell 参数 **HOME** 为缺省 *arg* 。参数 **PWD** 设置为当前目录。 Shell 参数 **CDPATH** 为包含 *arg* 的目录定义搜索路径。替换目录名称以冒号 (:) 分隔。如果 **CDPATH** 为空或未定义，缺省值为当前目录。请注意当前路径通过空路径名称指定，该名称紧跟在等号后面或在路径列表冒号分隔符之间的任何位置。如果 *arg* 以 **/** 开头，则不使用搜索路径。否则在每个路径下的目录搜索 *arg* 。另请参阅 *cd*(1) 。

第二种 **cd** 格式在当前目录名用字符串 *new* 替换字符串 *old* 及 **PWD** ，并试图将其改为新目录。

不能使用 **rksh** 执行 **cd** 命令。

echo [*arg* ...] 有关用法和说明，请参阅 *echo*(1) 。

% eval [*arg* ...] 读取参数，将其作为 Shell 的输入并执行得到的命令。

% **exec** [*arg* ...] 命令结束后参数分配功能保留。如果 *arg* 给定，由参数指定的命令在此 Shell 内执行，不创建新进程。输入/输出参数显示并影响当前进程。如果没有给定参数，此命令的效果为更改文件描述符，如输入/输出重定向列表所指定。在这种情况下，任何使用此机制打开的编号大于 2 的文件描述符在调用另一程序时关闭。

% **exit** [*n*] 导致 Shell 以 *n* 指定的状态退出。如果 *n* 省略，退出状态为最后执行命令的状态。文件末尾也导致 Shell 退出，除非 Shell 设置了 *ignoreeof* 选项（请参阅下文设置）。

%% **export** [*name* [**=***value*] ...]

给定 *name* 标记为自动导出至接下来执行命令的 *environment* 。

fc [**-e***ename*] [**-nlr**] [*first* [*last*]]

fc -e - [*old***=***new*] [*command*]

在第一种格式，从 *first* 到 *last* 范围内的命令从终端输入的最后 **HISTSIZE** 命令选择。参数 *first* 和 *last* 可以指定为数字或字符串。给定的字符串用于定位最近的命令。负数用于偏移当前命令编号。 **-l** 选项使命令在标准输出上列出。否则，编辑器程序 *ename* 在包含该键盘命令的文件上调用。如果不支持 *ename* ，参数 **FCEDIT** 的值（缺省值 **/usr/bin/ed** ）用作编辑器。编辑结束后，命令（如果存在）执行。如果 *last* 省略，只有 *first* 指定的命令被使用。如果 *first* 未指定，缺省值为先前命令时进行编辑，为 –16 时列表。 **-r** 选项保持命令顺序， **-n** 选项在列表时禁止命令编号。在后者中， *command* 在执行 *old***=***new* 替换之后重新执行。

fg [*job* ...] 按指定顺序将每个 *job* 引入前台。如果没有指定 *job* ，当前作业引入前台。有关 *job* 格式说明，请参阅作业。

getopts *optstring name* [*arg* ...]

检查 *arg* 的合法选项。如果省略了 *arg* ，则使用位置参数。使用一个 **+** 或一个 **-** 开头的选项参数。没有使用 **+** 或 **-** 开头的选项或者参数 **--** 结束选项。 *optstring* 包含 *getopts* 识别的字母。如果字母后面为 **:** ，则该选项需要一个参数。选项可以使用空白字符从参数中分隔出来。

当 *arg* 前面为 **+** 时，每次使用前置 **+** 调用变量 *name* ， **getopts** 放置从中查找到的下一选项字母。下一个 *arg* 的索引保存在 **OPTIND** 中。如果存在选项参数则保存在 **OPTARG** 中。

optstring 的前置 **:** 使 **getopts** 在 **OPTARG** 中存储无效选项的字母，并为 *name* 在未知选项时设置为 **?** ，在所需选项缺失时设置为 **:** 。否则， **getopts** 输出错误消息。没有更多选项时退出状态非零。另请参阅 *getopts*(1) 。

jobs [**-lnp**] [*job* ...]

如果 *job* 省略，则列出所有给定作业、或者活动作业的信息。除普通信息外， **-l** 还列出进程的 ID。 **-n** 选项仅显示在上次通知之后停止或退出的作业。 **-p** 选项仅列出进程组。有关 *job* 格式说明，请参阅作业。

kill [**-***sig*] *process* ...

将 TERM （终止）信号或指定信号发送到指定作业或进程。使用数字或名称给定信号（如 *signal*(5) 给定，去掉前缀 **SIG** ）。信号名称通过 **kill -l** 列出。无缺省值时，仅输入 **kill** 不影响当前

作业。如果发送 **TERM** （终止）或 **HUP** （挂起）信号，作业或进程在停止时发送 **CONT** （继续）信号。 *process* 参数可为进程 ID 或作业。如果 **kill** 的第一个参数为负整数，则会解释为 *sig* 参数而不是进程组。另请参阅 *kill*(1) 。

let *arg* ...
每个 *arg* 是单独的待计算的 “算术表达式”。有关算术表达式计算的说明，请参阅上文的算术计算。如果最后表达式的结果非零，退出状态为 0，否则为 1。

% newgrp [*arg* ...]
等效于 **exec newgrp** *arg* 。

print[**-Rnprsu**[*n*]] [*arg* ...]
Shell 输出机制。无选项或有 **-** 或 **--** 选项，参数在标准输出上输出，如 *echo*(1) 所述。原始模式， **-R** 或 **-r** 忽略 *echo* 的转义约定。 **-R** 选项输出除 **-n** 之外的所有后续参数和选项。 **-p** 选项使得参数写入到进程管道，该进程使用 **|&** 衍生替换标准输出。 **-s** 选项使参数写入到历史记录文件，而不是标准输出。 **-u** 选项可用于指定一位数字的文件分隔符单元编号 *n* ，替换输出。缺省值为 1。如果使用选项 **-n** ，不在输出中添加换行符。

pwd [**-L**|**-P**]
没有参数输出当前工作目录（等效于 **print -r - $PWD** ）。如果该当前目录为符号链接， **-L** 选项（缺省值）保留当前目录的逻辑含义， **-P** 保留当前目录的物理含义。请参阅特殊 **cd** 命令、 *cd*(1) 、 *ln*(1)) 和 *pwd*(1) 。

read [**-prsu**[*n*]] [*name*] [**?***prompt*] [*name* ...]
Shell 输入机制。读取一行并使用 **IFS** 作为分隔符将其拆分成单词。在 **-r** 原始模式，这一行末尾的 **** 并不表示该行继续。第一个单词分配给第一个 *name* ，第二个单词分配给第二个 *name* 等等，剩余单词分配给最后一个 *name* 。 **-p** 选项使用 **|&** 从 Shell 衍生的输入管道取出输入行。如果有 **-s** 选项，输入在历史记录文件中以命令形式保存。 **-u** 选项可用于指定一个一位文件描述符单元，用以读取。文件描述符可以使用 **exec** 特殊命令打开。 *n* 缺省值为 **0** 。如果 *name* 省略， **REPLY** 用作缺省 *name.* 。返回代码为 **0** ，除非到了文件末尾。文件末尾有 **-p** 选项时清除此进程，以便衍生另一进程。如果第一个参数包含 **?** ，当 Shell 为交互时，此单词的剩余部分用作提示。如果给定文件描述符为用于写入的打开状态并且是终端设备，则提示符置于此单元上。否则提示符发布在文件描述符 2 上。返回代码为 **0** ，除非到了文件末尾。另请参阅 *read*(1) 。

%% readonly [*name*[**=***value*] ...]
给定 *names* 标注只读并且不能被后续分配更改。

% return [*n*]
使得 Shell 函数返回调用脚本，返回状态由 *n* 指定。如果忽略 *n* ，返回状态为最后执行命令的状态。仅将 *n* 的低 8 位返回给调用者。如果 **return** 不是在 函数中或使用 **.** （点）内置命令执行脚本中被调用，具有与 **exit** 命令相同的效果。

set [±**aefhkmnopstuvx** | ±**o** *option*] **...** [±**A** *name*] [*arg* **...**]
下列选项用于该命令：

-A 组分配。未设置 *name* 和从列表 *arg* 按顺序分配值。如果使用 **+A** ，变量 *name* 并不先取消设置。

-a 所有后续定义参数自动导出。

-e 如果 Shell 不交互并且命令失败，如果已经设置过则执行 **ERR** 陷阱，则会立即退出。此模式在读取配置文件时禁用。

-f 禁用文件名生成。

-h 首次碰到名称为 标识符的命令时成为跟踪别名。

-k 所有参数分配参数（不仅仅是命令名称的前置）针对命令置于环境中。

-m 后台作业运行在单独进程组并在结束时输出一行。后台作业的退出状态在完成信息中报告。此选项在交互 Shell 自动打开。

-n 读取命令并检查语法错误，但不执行。交互 Shell 忽略 **-n** 选项。

-o **-o** 选项处理所有的多 *option* 名，但对每个 **-o** 只指定一个 *option* 。如果没有提供，则输出当前选项设置。 **-o** 参数 *option* 名称如下所示：

allexport	与 **-a** 相同。
bgnice	所有后台作业在较低优先级下运行。
errexit	与 **-e** 相同。
emacs	激活命令条目的 **emacs-**形式内置编辑器。
gmacs	激活命令条目的 **gmacs-**形式内置编辑器。
ignoreeof	Shell 在文件末尾不退出。必须使用 **exit** 命令。
keyword	与 **-k** 相同。
markdirs	所有来自文件名称生成的目录名称有一个追加结尾 **/** 。
monitor	与 **-m** 相同。
noclobber	防止从截取现存文件重定向 **>** 。启用时需要 **>|** 以截取文件。
noexec	与 **-n** 相同。
noglob	与 **-f** 相同。
nolog	不在历史记录文件保存函数定义。
nounset	与 **-u** 相同。
privileged	与 **-p** 相同。
verbose	与 **-v** 相同。
trackall	与 **-h** 相同。
vi	激活 **vi-**形式内置编辑器的的插入模式，直至按下 ESC 键，进入移动模式。返回发送该行。
viraw	将每个字符作为在 **vi** 模式下的输入处理。
xtrace	与 **-x** 相同。

-p 禁用 **$HOME/.profile** 文件处理并使用文件 **/etc/suid_profile** 替换 **ENV** 文件。此模式在有效 uid (gid) 与真实 uid (gid) 不同时启用。关闭此项使得有效 uid 和 gid 设置为真实 uid 和 gid。

-s 位置参数排序。

- **-t** 在读取和执行一个命令后退出。
- **-u** 在替换时将未设置参数作为错误处理。
- **-v** 在读取 Shell 输入行时输出。
- **-x** 在执行时输出命令及其参数。
- **-** 关闭 **-x** 和 **-v** 选项并停止检查选项参数。
- **--** 请勿更改任何选项；在设置 **$1** 的值以 **-** 开始时有用。如果此选项后无任何参数，则位置参数未设置。

在选项之前使用 **+** 替换 **-** ，使得该选项关闭。这些选项也可在调用 Shell 时使用。当前选项设置可以使用 **$-** 检查。

除非指定了 **-A** ，剩余 *arg* 参数为位置参数并连续分配为 **$1** 、 **$2** 等等。如果没有给定参数或选项，所有名称的值在标准输出上输出。

% shift [*n*] 从 **$***n***+1** ……重命名为 **$1** ……、缺省 *n* 为 1。参数 *n* 可以为任何算术表达式，计算小于或等于 **$#** 的非负数字。

test [*expr*] 计算位置表达式 *expr* 。有关用法和说明，请参阅 *test*(1) 。算术比较运算符不限定于整数。允许任何算术表达式。有四个附加基本表达式可用：

-L *file*	如果 *file* 是符号链接为真。
file1 **-nt** *file2*	如果 *file1* 比 *file2* 新则为真。
file1 **-ot** *file2*	如果 *file1* 比 *file2* 旧则为真。
file1 **-ef** *file2*	如果 *file1* 具有相同设备且信息节点号为 *file2* 时为真。

% times 输出 Shell 和从 Shell 中运行的进程的累积用户和系统时间。

% trap [*arg*] [*sig* ...]

arg 是一个命令，在 Shell 收到 *sig* 信号时运行（请注意 *arg* 在设置陷阱和使用陷阱的时候扫描一次）。每个 *sig* 可以给定为信号的编号或名称。陷阱命令按信号编号顺序执行。忽略任何试图在输入当前 Shell 中无效的信号上设置陷阱的操作。如果 *arg* 省略或为 **-** ，所有 *sig* 陷阱重置为原始值。如果 *arg* 为空字符串，此信号被 Shell 和调用命令忽略。如果 *sig* 为 **DEBUG** ，在每个命令之后执行 *arg* 。如果 *sig* 为 **ERR** ， *arg* 在命令退出代码非零的时候执行。如果 *sig* 为 **0** 或 **EXIT** 并且 **trap** 语句在函数内主体内执行，则命令 *arg* 在函数完成之后执行。如果 **trap** 设置于函数外部，其 *sig* 为 **0** 或 **EXIT** ，命令 *arg* 在退出 Shell 时执行。无参数的 **trap** 命令输出与每个信号编号关联的命令列表。

%% typeset [**±LRZfilrtux**[*n*]] [*name*[= *value*]] ...

命令完成后参数分配功能保留。如果在函数内部调用，创建参数 *name* 新的实例。参数值和类型在函数完成时存储 。可以指定下列列表的属性：

- **-L** 左对齐并从 *value* 删除开头空格。如果 *n* 非零，则定义字段宽度。否则，由第一个分配值的宽度确定。在 *name* 已经分配时，如有必要，该值右侧用空格填充或截断以适合字段。如果 **-Z** 选项也已设置，则删除开头的零。 **-R** 选项关闭。

-R 右对齐并在开头用空格填充。如果 *n* 非零则定义字段宽度。否则，该项由第一个分配值的宽度确定。如果重新分配参数，字段左填充空格或从末端截断。 **-L** 选项关闭。

-Z 如果第一个非空字符为数字且 **-L** 选项未设置，右对齐并用零填充开头。如果 *n* 非零，该项定义字段宽度。否则其宽度由第一个分配值确定。

-f 使得 *name* 引用函数名，而不是参数名。不能对使用 **typeset** 语句声明的 *name* 进行分配。此外仅有的有效选项是 **-t** （开启对此函数跟踪的执行）和 **-x** （允许在相同进程环境下，函数跨越 Shell 步骤保持功能）。

-i 参数为整数。使运算更快。如果 *n* 非零，则定义输出运算基数，否则首个分配确定输出基数。

-l 将所有大写字符转换为小写字符。大写 **-u** 选项关闭。

-r 任何给定 *name* 标记为“只读”并且不能被后续的分配改变。

-t 标记命名参数。标记可以由用户定义，对 Shell 没有特殊含义。

-u 将所有小写字符转换为大写字符。小写 **-l** 选项关闭。

-x 标记所有给定 *name* ，用于自动导出到后续执行命令的环境。

使用 **+** 替换 **-** 导致这些选项关闭。如果没有给定 *name* 参数，但指定了选项，输出设置了这些选项参数名称列表（其值可选）。使用 **+** 替换 **-** ，保留要输出的值。如果没有给定名称或选项，则输出所有参数的名称和属性。

ulimit [**-HSacdfst**] [*limit*]

设置或显示来源限制。指定来源的显示在指定 *limit* 时显示。在源指定的部分中 *limit* 的值可以为数字或关键字 **unlimited** 。

-H 和 **-S** 标志指定是否为给定源设置硬性限制 (-H) 或软性限制 (-S)。硬性限制在设置后不能增加。软性限制可以增加为硬性限制。如果没有指定 **-H** 或 **-S** ，则对两者均使用限制。

当前源限制在 *limit* 省略时输出。这种情况下，输出软性限制，除非指定 **-H** 。指定多个源的时候，在输出值之前输出名称和单元。

如果没有给定选项，假定为 **-f** 。

-a 列出当前全部源限制。

-c 列出或设置核心转存块大小为 512 字节。

-d 列出或设置数据区域大小的千字节数。

-f 列出或设置子进程写入文件块为 512 字节（可读取的文件大小）。

-s 列出或设置堆栈区域大小的千字节数。

-t 列出或设置每个进程使用的秒数。

umask [*mask*] 用户文件创建掩码设置为 *mask* （请参阅 *umask*(2) ）。 *mask* 可以为八进制数或符号值，如 *chmod*(1) 所述。如果符号值给定，新的 umask 值为应用 *mask* 至先前 umask 值的补码所得到的补码。如果省略 *mask* ，输出当前掩码的值。另请参阅 *umask*(1) 。

unalias *name* ...

由 *name* 列表给出的参数从 *alias* 列表删除。

unset [**-f**] *name* ...

由 *name* 给出的列表参数未分配，即它们的值和属性被清除。不能取消只读变量的设置。如果设置了 **-f** 选项，则 *name*s 引用功能名。取消设置 **ERRNO**、 **LINENO**、 **MAILCHECK**、 **OPTARG**、 **OPTIND**、 **RANDOM**、 **SECONDS**、 **TMOUT**、 _ 删除它们的特殊含义，即使它们接下来被设置。

% wait [*job*]

等待指定 *job* 以终止或停止，并报告其状态。此状态成为返回代码，用于 **wait** 命令。如果未给定 *job* ，则 **wait** 等待全部当前活动子进程的终止或停止。返回的终止状态为最后进程状态。有关 *job* 的格式说明，请参阅作业。

whence [**-pv**] *name* ...

对每个 *name* ，如果作为命令名称使用则指示其解释方式。 **-v** 选项生成一个更加详细的报告。 **-p** 选项为 *name* 进行路径搜索，即使 *name* 是别名、函数或者保留词。

调用 ksh

如果 **exec** 调用 Shell（请参阅 *exec*(2)），参数零的第一个字符 ($0) 为 **-** 。Shell 假定为登录 Shell，命令首先从 **/etc/profile** 读取。表达式 **${HOME:-.}/.profile** 随后计算并尝试打开得到的文件名。如果文件名成功打开则读取文件。下一步，如果文件存在，命令通过为环境参数 **ENV** 的值执行参数替换，从文件命名中读取命令。如果未设置 **-s** 选项，且设置了 *arg* ，则第一个 *arg* 执行路径搜索以确定执行的脚本名称。具有 *arg* 的 **ksh** 运行时，脚本 *arg* 必须具有读取权限，忽略所有 *setuid* 和 *getgid* 设置。令名按如下所述读取。下列选项在调用时由 Shell 解释：

-c *string* 如果存在 **-c** 选项，则从 *string* 读取命令。

-s 如果存在 **-s** 选项或者没有保留参数，从标准输入读取命令。 Shell 输出，除了上文列举的部分特殊命令，写入到文件描述符 2。

-i 如果存在 **-i** 选项或 Shell 输入和输出附加指定到终端， Shell 为交互式。在这种情况下，忽略 SIGTERM（这样， **kill 0** 不终止交互式 Shell），SIGINT +1 被捕获和忽略（这样， **wait** 可中断）。在所有情况下，Shell 忽略 SIGQUIT。（请参阅 *signal*(5)）。

-r 如果设置了 **-r** 选项，则 Shell 为受限 Shell。

其余选项和参数的描述请参阅上文的 **set** 命令。

仅 rksh

rksh 用于为功能比标准环境下更受限的 Shell 设置登录名和执行环境。 **rksh** 的行为与 **ksh** 相同，除了下列禁止项：

- 更改目录（请参阅 *cd*(1)）
- 设置 **SHELL**、 **ENV** 或 **PATH** 的值
- 指定包含 **/** 的路径或命令名。

- 重定向输出（ **>** 、 **>|** 、 **<>** 和 **>>** ）

上述限制在 **.profile** 和 **ENV** 文件解释之后生效。

当待执行命令为 Shell 步骤时， **rksh** 调用 **ksh** 以执行。因此，为最终用户提供可访问标准 Shell 的全部功能的 Shell 步骤，同时受限于命令限制菜单。此方案假定最终用户在同一目录下没有写入和执行权限。

当 **rksh** 调用 Shell 过程时，使用 **#!** 幻数指定的 Shell 解释程序继承 **rksh** 的所有受限功能。因此，为了在 **rksh** 下执行、期望使用全部标准 Shell 功能的 Shell 过程不能指定具有 **#!** 的解释程序。

通过这些规则执行可靠的设置行为并保持用户在合适的路径（不一定为登录路径），有效地为 **.profile** 编写者提供了对用户行为的完全控制。

系统管理员经常为命令设置一个目录（通常为 **/usr/rbin** ），可以通过 **rksh** 安全调用。HP-UX 系统提供了适用于受限用户的受限编辑器 **red** （请参阅 *ed*(1) ）。

命令行编辑

内置编辑选项

通常，在终端输入的每个命令行后面跟随换行符（回车或换行）。如果设置了 **emacs** 、 **gmacs** 或 **vi** 选项之一，则用户可以编辑命令行。当 **VISUAL** 或 **EDITOR** 变量分配了以这些选项名称结束的值时，自动选择编辑选项。

编辑功能需要用户终端接受 **Return** 作为无换行的回车，并且空格字符将覆盖屏幕上的现有字符。 ADM 终端用户应将“space/advance”切换为“space”。 HP 终端用户应将 straps 设置为“bcGHxZ etX”。

编辑模式允许用户在窗口中监视当前行。缺省窗口宽度为 80，除非定义了 **COLUMNS** 值。如果一行比窗口宽度减二要长，窗口末端显示记号以提醒用户。该记号为 **>** 、 **<** 或 ***** 分别对应着行的右侧、左侧或两端的窗口。光标在窗口中移动且可抵达边界，窗口光标居中。

每个编辑模式下的搜索命令提供对历史记录文件的访问。仅有字符串匹配，没有模式，虽然字符串开始的 ^ 限制从行的第一个字符开始匹配。

Emacs 编辑模式

此模式可以通过 **emacs** 或 **gmacs** 选项调用。它们的区别仅仅在于处理 **^T** 。编辑时用户移动光标到需要校正的位置，然后插入或删除字符或单词。所有编辑命令均为控制字符或转义序列。控制字符的符号为插字字符 (^) 后跟随字符。例如， **^F** 为 Ctrl-**F** 的符号。按 **f** 键的同时按住 Ctrl (control) 键输入。请 不要按下 Shift 键。（符号 **^?** 指示 DEL (delete) 键）。

转义序列的符号为 **M-** ，后跟随字符。例如， **M-f** （发音 Meta f）通过按下 ESC (ASCII 033) 后跟随 **f** 。 **M-F** 符号为 ESC，后跟随 Shift（大写） **F** 。

所有编辑命令从行的任意位置操作（不仅仅是开始）。 **Return** 和 **Line Feed** 键在编辑命令之后都不输入，除非被提及。

^F	光标向前（右）移动一个字符。
M-f	光标向前移动一个单词。（编辑器的一个单词的含义为一个仅包含字母、数字和下划线的字符串）。

^B	光标向后（左）一个字符。
M-b	光标向后移动一个单词。
^A	光标移动到本行开始。
^E	光标移动到本行结尾。
^]*char*	光标向前移动到本行的 *char* 字符。
M-^]*char*	光标向后移动到本行的 *char* 字符。
^X^X	交换光标和标记。
erase	（用户使用 *stty*(1) 命令定义的清除字符，通常为 **^H** 或 **#** ）。删除前一字符。
^D	删除当前字符。
eof	文件末尾字符，通常为 **^D** ，如果当前行为空则终止 Shell。
M-d	删除当前单词。
M-^H	（Meta-删除键）删除前一单词。
M-h	删除前一单词。
M-^?	(Meta-DEL) 删除前一单词。如果中断字符为 **^?** （缺省为 DEL），此命令无效。
^T	以 **emacs** 模式调换当前字符和下一个字符。以 **gmacs** 模式调换前两个字符。
^C	大写当前字符。
M-c	大写当前单词。
M-l	将当前单词改为小写。
^K	从光标处删除至行的结尾。如果前面为数字参数，其值小于当前光标位置，从给定位置删除至光标处。如果前面为数字参数，其值大于光标当前位置，从光标处删除至给定位置。
^W	从光标处清除至记号。
M-p	将光标至记号内的区域放入堆栈。
kill	（用户定义的清除字符，使用 *stty*(1) 命令定义，通常为 **^G** 或 **@** ）。清除当前行。如果连续输入两个 *kill* 字符，所有接下来的 kill 字符导致换行。（使用纸张中断时有用）。
^Y	恢复上次从行中删除的项目（将移出项目返回行）。
^L	换行并输出当前行。
@	（空字符）设置记号。
M-空格字符	（Meta 空间）设置记号。
^J	（换行）执行当前行。
^M	（回车）执行当前行。
^P	提取前一命令。每次输入 **^P** 时，访问历史记录列表中的下一个先前命令。
^N	提取下一命令。每次输入 **^N** 命令，访问历史记录列表中的下一个命令。
M-<	提取最早的历史记录行。
M->	提取最新的历史记录行。
^R*string*	逆向搜索历史记录中的包含 *string* 的先前命令行。如果参数给定为零，则向前搜索。 *string* 由 **Return** 或 **Newline** 终止。如果 *string* 前面为 ^ ，匹配行必须以 *string* 开始。如果 *string* 省略，访问最近包含 *string* 的下一命令行。在这种情况下，零参数更改搜索的方向。
^O	操作 - 执行当前行并从历史记录文件提取与当前行相关的下一行。

M-*digits* 定义数字参数。数字作为下一个命令的参数。命令接受的参数包括 **^F** 、 **^B** 、 *erase* 、 **^C** 、 **^D** 、 **^K** 、 **^R** 、 **^P** 、 **^N** 、 **^]** 、 **M-.** 、 **M-_** 、 **M-b** 、 **M-c** 、 **M-d** 、 **M-f** 、 **M-h** 、 **M-l** 和 **M-^H** 。

M-*letter* 功能键。用户的别名列表使用名称 *_letter* 搜索别名，如果定义了此名称的别名，其值插入输入队列。此 *letter* 一定不为上述元函数。

M-. 先前命令的最后一个单词插入行中。如果前面为数字参数，此参数的值确定插入的单词，而不是最后一个单词。

M-_ 与 **M-.** 相同。

M-* 在当前单词上尝试文件名生成。

M-ESC 完成文件名。使用与当前单词匹配的所有的文件名中最长的通用前缀替换当前单词，其后追加星号。如果匹配唯一，文件若为目录则追加 **/** ，若不为目录则追加空格。

M-= 列出匹配当前单词模式的文件，视为其后追加了星号。

^U 下一命令的参数乘以 4。

**** 删除下一字符。编辑字符，如果前面为 **** ，用户的清除、终止和中断（通常为 **^?** ）字符可以在命令行输入或在搜索字符串中。 **** 删除下一字符的编辑功能（如果存在）。

^V 显示 Shell 版本。

M-# 在行的开头插入 **#** 并执行。这会在历史记录文件中插入注释。

Vi 编辑模式

有两种输入模式。输入命令使用户进入 输入模式。要进行编辑，用户按 ESC 键进入 控制模式，移动光标至需要校正的位置，然后插入或删除字符或单词。大部分控制命令接受命令之前可选的重复 *count* 。

大部分系统的在 vi 模式下，从开始即启用规范处理，如果速度大于等于 1200 波特率并且包含任意控制字符，命令再次回显，或从提示输出以后时间不足一秒。 ESC 字符终止命令剩余部分的规范处理，用户接下来可以修改命令行。此方案具有规范处理的优点，使用原始模式的预先键入回显。

设置 **viraw** 选项始终禁用终端上的规范处理。此模式对系统隐藏，不支持两个备用文件末尾分隔符，这对于此类终端很有用。

输入编辑命令

编辑器的缺省值为输入模式。

erase 删除先前字符（ erase 为用户定义的清除字符，使用 *stty*(1) 命令定义，通常为 **^H** 或 **#** ）。

^W 删除上一个空格分隔单词。

^D 终止 Shell。

^V 删除下一个字符。编辑字符，清除或终止字符可以在命令行或前面为 **^V** 的搜索字符串输入。 **^V** 删除下一个字符的编辑功能（如果存在）。

**** 转义下一个 *erase* 或 *kill* 字符。

移动编辑命令

这些命令移动光标。指定 [count] 使命令重复引用。

[*count*]**l** 光标向前（右）一个字符。

[*count*]**w** 光标向前一个字母数字单词。

[*count*]**W** 光标到后面为空格的下一个单词开头。

[*count*]**e** 光标移动到单词的结尾。

[*count*]**E** 光标移动到当前空白界定的单词。

[*count*]**h** 光标向后（左）一个字符。

[*count*]**b** 光标向后一个单词。

[*count*]**B** 光标移动到前面有空白字符分隔的单词。

[*count*]**|** 光标移动到列 *count* 。缺省值为 1。

[*count*]**f***c* 在当前行查找下一个字符 *c* 。

[*count*]**F***c* 在当前行查找先前字符 *c* 。

[*count*]**t***c* 等效于 **f** ，后面为 **h** 。

[*count*]**T***c* 等效于 **F** ，后面为 **l** 。

[*count*]**;** 重复上一单个字符查找命令 **f** 、 **F** 、 **t** 或 **T** 。

[*count*]**,** 将上一个单个字符查找命令逆向。

0 光标移动到行的开始。

^ 光标移动到行的第一个非空字符。

$ 光标移动到行的结尾。

搜索编辑命令

这些命令访问用户命令历史记录。

[*count*]**k** 提取先前命令。每次按下 **k** ，则访问历史记录列表中的下一个前期命令。

[*count*]**-** 等效于 **k** 。

[*count*]**j** 提取下一个命令。每次输入 **j** ，访问历史记录中的下一个后面的命令。

[*count*]**+** 等效于 **j** 。

[*count*]**G** 提取命令编号 *count* 。缺省为历史记录菜单的第一个命令。

/*string* 在历史记录中向后搜索包含 *string* 的先前命令。 *string* 被 **Return** 或 **Newline** 终止。如果 *string* 前面为 ^ ，匹配行必须以 *string* 开始。如果 *string* 为空，则使用先前字符串。

?*string* 与 **/** 相同，但向前搜索。

n 搜索与最后模式相匹配的下一个 **/** 或 **?** 命令。

N 搜索与最后模式相匹配的下一个 **/** 或 **?** ，但方向相反。在历史记录中搜索通过先前 **/** 命令输入的 *string* 。

文本修改编辑命令

这些命令修改行。

a 在当前字符后进入输入模式并输入文本。

A 将文本追加到行的结尾。等效于 **$a** 。

[*count*]**c***motion*

c[*count*]*motion* 移动光标至由 *motion* 指定的字符位置，删除原始光标位置和新位置之间的全部字符，进入输入模式。如果 *motion* 为 **c** ，删除整行并进入输入模式。

C 删除直至行结尾的当前字符，并进入输入模式。等效于 **c$** 。

S 等效于 **cc** 。

D 删除直至行结尾的当前字符。等效于 **d$** 。

[*count*]**d***motion*

d[*count*]*motion* 移动光标至由 *motion* 指定的字符位置，删除原始光标位置和新位置之间的全部字符。如果 *motion* 为 **d** ，删除整行。

i 进入输入模式，并在当前字符之前插入文本。

I 在行开头之前插入文本。等效于两个字符序列 **0i** 。

[*count*]**P** 在光标前置入先前文本修改。

[*count*]**p** 在光标之后置入先前文本修改。

R 进入输入模式并使用用户覆盖输入方式替换屏幕上的字符。

[*count*]**r***c* 使用 *c* 更换当前字符。

[*count*]**x** 删除当前字符。

[*count*]**X** 删除前置字符。

[*count*]**.** 重复先前文本修改命令。

[*count*]**~** 转换当前字符的大小写，并向前移动光标。

[*count*]**_** 先前命令的 *count* 单词追加在当前光标位置并在追加文本的结尾置编辑器于输入模式。如果省略 *count* 则使用最后一个单词。

***** 追加 ***** 至当前单词并尝试文件名生成。如果未找到匹配则响铃。如果找到匹配，单词被匹配字符串替换，命令置编辑器于输入模式。

ESC

**** 尝试在当前单词完成文件名。使用与当前单词匹配的所有文件名中最长的通用前缀替换当前单词，其后追加星号。如果匹配唯一，文件若为目录则追加 **/** ，若不为目录则追加空格。

其他编辑命令

[*count*]**y***motion*

y[*count*]*motion* 通过字符移出当前字符， *motion* 将移动光标并将字符放入删除缓冲区。文本和光标保持不变。

Y 从当前位置移动到行结尾。等效于 **y$** 。

u 恢复最后文本修改命令。

U 恢复所有在行上执行的文本修改命令。

[*count*]**v** 返回在输入缓冲区的命令 **fc -e ${VISUAL:-${EDITOR:-vi}}** *count* 。如果省略了 *count*，则使用当前行。

^L 换行并输出当前行。仅在控制模式下有效。

^J （换行）执行当前行，忽略模式。

^M （回车）执行当前行，忽略模式。

等效于 **I#** ，后面为 **Return** 。在每个行前插入 **#** 后和每个换行符之后发送行。在历史记录列表中在插入当前命令行而不执行时有用。

= 如果其后追加了星号，列出匹配当前单词的文件名。

@*letter* 在用户的别名列表中使用名称 _*letter* 搜索别名，如果此名称定义了别名，其值插入输入队列以处理。

外部语言环境影响

环境变量

LC_COLLATE 可确定为生成文件名评估模式匹配表示法时使用的排序序列。

LC_CTYPE 确定字符以字母分类，和在模式匹配表示法中与字符分类表达式匹配的字符。

如果未在环境中指定 **LC_COLLATE** 或 **LC_CTYPE** ，或者将其设置为空字符串，则 **LANG** 的值将用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则使用缺省的“C”（请参阅 *lang*(5)）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **ksh** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

KSH_QUOTEMC 将引用的元字符转换为以 "[[*string* = *pattern*]]" 结构处理。如果 **KSH_QUOTEMC=true** 在环

境中定义，则 *pattern* 的任何部分均可以引用以使其作为字符串匹配。这种用法遵循惯例 *dtksh*(1) 。如果在环境中未定义 **KSH_QUOTEMC**，则处理遵循传统 Korn Shell 管理。

国际代码集支持

支持单字节字符代码集。

返回值

Shell 检测到错误，如语法错误，将使得 Shell 返回非零退出状态。否则，Shell 返回最后执行命令的退出状态（另请参阅上文的 **exit** 命令）。如果 Shell 为非交互性使用，放弃执行 Shell 文件。由 Shell 检测到的运行错误通过输出命令或函数名称和错误状态进行报告。如果发生错误的行编号大于一，则行编号在命令或函数名称之后输出于方括号 ([ ]) 内。

警告

文件描述符 10 和 54 至 60 被 Korn Shell 内部使用。使用这些描述符的应用程序和派生子 Shell 不应取决于它们在子 Shell 或 子代的存在。

如果执行了 跟踪别名的任务，并且搜索路径的目录中安装了相同名称的命令，找到原始命令存在的目录之前，Shell 继续加载和执行原始命令。使用 **alias** 命令 的 **-t** 选项修正此状态。

如果用于移动当前目录或上层目录， **pwd** 可能不会正确响应。使用全路径名称的 **cd** 命令修正此状态。

某些非常旧的 Shell 脚本包含插入记号 (ˆ)，用作管道字符 (|) 的同义名称。请注意， **ksh** 不将插入字符识别为管道字符。

如果一个命令传送至 Shell 命令，Shell 命令中的所有变量在命令完成后丢失。

在复合命令中使用 **fc** 内置命令导致全部命令从历史记录文件中消失。

内置命令 **.** 文件在命令执行之前读取全部文件。因此，文件中的 **alias** 和 **unalias** 命令不应用到任何文件定义的函数。

在 Sehll 等待前台作业的时候不处理陷阱。因此， **CHLD** 上的陷阱在前台作业终止前不执行。

export 内置命令不能正确处理组。只有组的第一个元素导出至 *environment* 。

从非交互式 Shell 开始的后台处理不能使用作业控制命令访问。

在国际化环境中，排序序列的字符顺序由 **LC_COLLATE** 设置确定，而不是字符值的二进制顺序。这将带来一定的危险，尤其是在文件名生成模式下使用范围表达式。例如，命令：

rm [a-z]*

希望匹配所有使用小写字母字符开头的文件名称。但是，如果由 **LC_COLLATE** 指定字典顺序，也可能匹配以大写字符开头的文件名称（如同那些使用着重字母的开头）。相反，在诸如丹麦语或挪威语中，在 **z** 之后的字母排序匹配可能失败。

在国际环境中正确（同时也安全）的匹配特定字符分类的方法是使用模式的格式：

rm [[:lower:]]*

对所有支持的语言和代码集使用 **LC_CTYPE**

以确定字符分类和可预知工作。对于在非国际化系统上生成的 Shell 脚本（或者没有考虑上述危险），推荐在非 NLS 环境下执行。这需要将 **LANG**、 **LC_COLLATE** 等设置为“C” 或完全不设置。

请注意 **IFS** 变量的值在用户环境中影响脚本的行为。

ksh 通过在命令和自身之间创建管道完成命令替换。如果根文件系统已满，后续命令不能写入到管道。作为结果，Shell 不从命令接收输入，替换的结果为空。尤其是在这种环境下为参数分配使用命令替换将使得参数被以静默方式分配 NULL 值。

here-documents 的内容存储在名为 **/tmp/sh***pid***.***number* 的文件中。结束使用后删除临时文件时请注意。但是，由于设计限制，部分临时文件无法删除。

作者

ksh 由 AT&T 开发。

文件

/etc/passwd	用于查找主目录。
/etc/profile	读取以设置系统环境。
/etc/suid_profile	安全配置文件
$HOME/.profile	读取以设置用户自定义环境
/tmp/sh*	用于本文档

另请参阅

cat(1)、 cd(1)、 echo(1)、 env(1)、 getopts(1)、 kill(1)、 pwd(1)、 read(1)、 test(1)、 time(1)、 umask(1)、 vi(1)、 dup(2)、 exec(2)、 fork(2)、 gtty(2)、 pipe(2)、 stty(2)、 umask(2)、 ulimit(2)、 wait(2)、 rand(3C)、 a.out(4)、 profile(4)、 environ(5)、 lang(5)、 regexp(5)、 signal(5)。

名称

ktutil - Kerberos keytab 文件维护实用程序

概要

ktutil

说明

ktutil 命令调用一个 Subshell，管理员可以通过它读取、写入或编辑 Kerberos V5 keytab 或 V4 srvtab 文件中的条目。

ktutil 命令

list	显示当前密钥列表。别名： **l**
read_kt *keytab_filename*	将 Kerberos V5 keytab 文件 *keytab_filename* 读入当前密钥列表中。别名： **rkt**
read_st *srvtab_filename*	将 Kerberos V4 srvtab 文件 *srvtab_filename* 读入当前密钥列表中。别名： **rst**
write_kt *keytab_filename*	将当前密钥列表写入 Kerberos V5 keytab 文件 *keytab_filename* 中。别名： **wkt**
write_st *srvtab_filename*	将当前密钥列表写入 Kerberos V4 srvtab 文件 *srvtab_filename* 中。别名： **wst**
clear_list	清除当前密钥列表。别名： **clear**
delete_entry *slot*	从当前密钥列表中删除编号为 *slot* 的插槽中的条目。别名： **delete**
list_requests	显示可用命令列表。别名： **lr** 、 **?**
quit	从 **ktutil** 中退出。别名： **exit** 、 **q**

作者

ktutil 由麻省理工学院开发。

文件

/etc/krb5.keytab	keytab 文件的缺省位置。
/etc/srvtab	srvtab 文件的缺省位置

另请参阅

kerberos(5)。

名称

kvno - 输出 Kerberos 主体的密钥版本号

概要

kvno [**-e** *etype*] *service1* , [*service2*, ...]

说明

kvno 获取指定 Kerberos 主体的服务凭证，并输出每个主体的密钥版本号。

选项

-e *etype* 指定为命令行上命名的所有服务的会话密钥所请求的加密类型。在某些向后兼容性的情况下，该选项很有用。加密类型的值可以是 DES-CBC-CRC、DES-CBC-RAW 或 DES-CBC-MD4 之一。

service1,*service2* 一个或多个服务名或主体名。

环境变量

kvno 使用以下环境变量:

KRB5CCNAME
凭证缓存的位置。

作者

kvno 由 FundsXpress, INC. 开发。

文件

/tmp/krb5cc_{*uid*} 凭证缓存的缺省位置。{*uid*} 是用户的十进制 UID。

另请参阅

kdestroy(1)、 kinit(1)、 libkrb5(3)、 krb5.conf(4)、 kerberos(5)。

名称

last、lastb - 指明用户和 tty 的上一次登录

概要

/usr/bin/last [**-R**] [**-***number*] [**-x**] [**-X**] [**-f** *file*] [*name ...*] [*tty ...*]

/usr/bin/lastb [**-R**] [**-***number*] [**-x**] [**-X**] [**-f** *file*] [*name ...*] [*tty ...*]

说明

last 命令向后搜索 **/var/adm/wtmps** 文件（其中包含所有登录和注销的记录），以获得关于用户、tty 或用户和 tty 的任何组的信息。参数指定有意义的用户或 tty 的名称。可以给出 tty 名称的完整形式或缩写形式。例如， **last 0** 与 **last tty0** 相同。如果给出多个参数，则输出应用于任何参数的信息。例如， **last root console** 列出所有 **root** 的会话，以及控制台终端的所有会话。 **last** 命令首先输出最新的指定用户和 tty 的会话，指明会话开始的时间、会话持续的时间以及进行会话的 tty。 **last** 指明会话是否仍在进行，或者是否被重新引导中断。

每次系统重新引导时，伪用户 **reboot** 都要进行记录。因此， **last reboot** 命令在计算系统重新引导之间的相对时间方面是非常有用的。

如果中断 **last** ，则它会在 **wtmp** 中指明搜索的进展情况。如果被退出信号（该信号由 Ctrl-\ 生成）中断，则 **last** 会指明搜索的进展情况，然后继续搜索。

lastb 命令向后搜索数据库文件 **/var/adm/btmps** ，以显示错误登录信息。对 **/var/adm/btmps** 的访问应该只局限于具有相应权限的用户（只有 **root** 才可以拥有和读取），因为其中可能包含口令信息。

选项

last 命令和 **lastb** 命令可识别下列选项和参数。

(none) 如果未指定参数，则 **last** 按照相反顺序输出所有登录和注销的记录，首先输出最新的记录。

-R 与 **last** 和 **lastb** 一起使用时， **-R** 分别显示存储在 **/var/adm/wtmps** 和 **/var/adm/btmps** 中的用户主机名。主机名显示在 tty 名称和用户登录时间之间。

-*number* 将报告的行数限制为 *number* 。

-f *file* 使用 *file* 作为记帐文件的名称，而不是使用 **/var/adm/wtmp** 或 **/var/adm/btmp** 。

-X 使用 *file* 作为记帐数据库的名称，而不是使用 **/var/adm/wtmps** 。此选项应与 **-f** *file* 选项一起使用。

-x 如果此标志与 **-X** 标志一起使用，则以长格式显示各字段。在不使用 **-X** 标志的情况下，显示正常输出。

作者

last 由加州大学伯克利分校和 HP 联合开发。

文件

/var/adm/btmp 错误登录数据库

/var/adm/wtmp	登录数据库
/var/adm/wtmps	新建登录数据库
/var/adm/btmps	新建错误登录数据库

另请参阅

login(1)、 utmp(4)、 wtmps(4)。

名称

lastcomm - 反序显示以前执行的命令

概要

lastcomm [*commandname*] ... [*username*] ... [*terminalname*] ...

说明

lastcomm 输出有关以前执行的命令的信息。如果不指定任何参数，则 **lastcomm** 会输出在当前记账文件生命周期中记录在记账文件 **/var/adm/pacct** 中的所有命令的相关信息。如果指定了参数，则仅输出与命令名、用户名或终端名匹配的记账条目。例如，要生成用户 **root** 在终端 **ttyd0** 上执行的 **a.out** 命令的完整列表，可使用：

lastcomm a.out root ttyd0

对于每一个进程条目，将输出以下信息。

- 运行进程的用户名。
- 由系统中的记账工具收集的标志。
- 调用进程的命令名。
- 进程耗用的 CPU 时间（秒）。
- 进程启动的时间。

各标志的说明如下：

S 由拥有适当权限的用户执行命令。

F 在派生后运行命令，但接下来不会调用 *exec* 。

D 命令结束，同时生成 **core** 文件。

X 命令终止时发出 SIGTERM 信号。

文件

/var/adm/pacct 每个进程的当前记账文件

作者

lastcomm 由加州大学伯克利分校开发。

另请参阅

last(1)、 acct(4)、 acctsh(1M)、 core(4)。

名称

ld - 链接编辑器

概要

备注

对于基于 Itanium® 的系统，请参阅 *ld_ia*(1) 。

对于 PA-RISC 系统，请参阅 *ld_pa*(1) 。

使用 **uname** 命令确定您的系统类型。在基于 Itanium 的系统上， **uname -m** 返回 **ia64** 。所有其他值表示 PA-RISC 系统。

另请参阅

ld_ia(1)、 ld_pa(1)、 uname(1)。

名称

ldd - 列出可执行文件或共享库的动态相关性

概要

备注

对于基于 Itanium® 的系统，请参阅 *ldd_ia*(1) 。

对于 PA-RISC 系统，请参阅 *ldd_pa*(1) 。

使用 **uname** 命令确定您的系统类型。在基于 Itanium 的系统上， **uname -m** 返回 **ia64** 。所有其他值表示 PA-RISC 系统。

另请参阅

ldd_ia(1)、 ldd_pa(1)、 uname(1)。

名称

ldd_ia： ldd - 列出可执行文件或共享库的动态相关性

概要

ldd [**-d**] [**-r**] [**-s**] [**-v**] *filename*...

说明

ldd 命令可以列出不完整可执行文件或共享库的动态相关性。

ldd 列出有关动态相关性和符号引用的详细信息。如果对象文件是可执行文件， **ldd** 会列出将作为执行该文件的结果而加载的所有共享库。如果该文件是共享库， **ldd** 会列出将作为加载该库的结果而加载的所有共享库。

ldd 使用与动态加载程序（ /usr/lib/hpux32/dld.so 和 **/usr/lib/hpux64/dld.so** ）相同的算法在运行时定位共享库。有关详细信息，请参阅 *dld.so*(5) 中的“动态路径列表”。

选项

ldd 可识别下列选项：

- **-d** 检查对数据符号的引用。
- **-r** 检查对数据和代码符号的引用。
- **-s** 显示用于定位共享库的搜索路径。
- **-v** 显示所有相关性关系。

外部语言环境影响

环境变量

ldd 使用以下环境变量来定位共享库。

LD_LIBRARY_PATH
在运行时定义共享库搜索路径的路径名的冒号分隔列表。有关详细信息，请参阅 *dld.so*(5) 中的“动态路径列表”。

SHLIB_PATH
在运行时定义共享库搜索路径的路径名的冒号分隔列表。有关详细信息，请参阅 *dld.so*(5) 中的“动态路径列表”。

下列国际化变量会影响 **ldd** 的执行：

LANG 当未提供 **LC_ALL** 和其他 **LC_*** 环境变量时，确定本国语言、本地惯例和编码字符集的语言环境类别。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省的 **C** （请参阅 *lang*(5) ）而非 **LANG** 。

LC_ALL
确定所有语言环境类别的值，其优先级高于 **LANG** 和其他 **LC_*** 环境变量。

LC_MESSAGES
确定语言环境，用于影响写入标准错误中的诊断消息的格式与内容。

LC_NUMERIC

确定数字格式化时所用的语言环境类别。

LC_CTYPE

确定字符处理功能的语言环境类别。

NLSPATH

为处理 **LC_MESSAGES** 确定消息目录的位置。

如果任一国际化变量包含无效设置，则 **ldd** 就会认为所有国际化变量均设置为 **C** 。请参阅 *environ*(5) 。

诊断信息

ldd 将共享库路径名的记录输出到 stdout。可选的符号解析问题列表将输出到 stderr。

ldd 在操作成功时返回零。非零的返回代码指示已出错。

举例

缺省情况下， **ldd** 将输出简单的动态路径信息。它由可执行文件（或共享库）中记录的相关性后接这些库所在的物理位置组成。

```
ldd a.out
  ./libx.so =>    ./libx.so
  libc.so =>      /usr/lib/hpux32/libc.so.1
  libdl.so =>     /usr/lib/hpux32/libdl.so.1
```

-v 选项使得 **ldd** 在输出动态路径信息的同时输出相关性关系。

```
ldd -v a.out
  find library=./libx.so; required by a.out
    ./libx.so =>    ./libx.so
  find library=libc.so; required by a.out
    libc.so =>      /usr/lib/hpux32/libc.so.1
  find library=libdl.so; required by /usr/lib/hpux32/libc.so.1
    libdl.so =>     /usr/lib/hpux32/libdl.so..1
```

ldd 的 **-r** 选项将使其分析所有符号引用，并输出有关不满足的代码和数据符号的信息。

```
ldd -r a.out
    ./libx.so =>    ./libx.so
    libc.so =>      /usr/lib/hpux32/libc.so.1
    libdl.so =>     /usr/lib/hpux32/libdl.so.1
    symbol not found: val1 (./libx.so)
    symbol not found: count (./libx.so)
    symbol not found: func1 (./libx.so)
```

警告

ldd 不会列出用 *dlopen*(3C) 或 *shl_load*(3X) 显式加载的共享库。

文件

a.out	输出文件
/usr/lib/hpux32/dld.so	基于 32 位 Itanium(R) 的系统动态加载程序
/usr/lib/hpux64/dld.so	基于 64 位 Itanium 的系统动态加载程序
/usr/ccs/lib/hpux32/lddstub	32 位虚拟可执行文件，加载来检查共享库的相关性
/usr/ccs/lib/hpux64/lddstub	64 位虚拟可执行文件，加载来检查共享库的相关性
/usr/lib/nls/$LANG/ldd.cat	消息清单

另请参阅

系统工具

ld(1) 调用链接编辑器

其他信息

a.out(4) 汇编程序、编译程序和链接程序输出
dld.so(5) 动态加载程序

文本和教程

《HP-UX Linker and Libraries User’s Guide》

名称

ldd_pa： ldd - 列出可执行文件或共享库的动态相关性

概要

ldd [**-b**] [**-d**] [**-r**] [**-s**] [**-v**] *filename*...

说明

ldd 是一个可以列出不完整可执行文件或共享库的动态相关性的命令。

ldd 列出有关动态相关性和符号引用的详细信息。如果对象文件是可执行文件，则 **ldd** 列出将作为执行该文件的结果加载的所有共享库。如果它是共享库，则 **ldd** 列出将作为加载该库的结果加载的所有共享库。

ldd 使用与动态加载程序（ /usr/lib/dld.sl 和 **/usr/lib/pa20_64/dld.sl** ）相同的算法定位共享库。有关详细信息，请参阅 *dld.sl*(5) 中的下列各节： PA-RISC 32 位动态路径列表、 PA-RISC 64 位动态路径列表和 LD_PRELOAD 环境变量。

选项

ldd 可识别下列选项：

-b 仅适用于 *PA-RISC 32* 位。与 **-d** 和（或） **-r** 联合使用，以强制 **dld.sl** 绑定所有相关库和报告无效情况。缺省情况下， **dld.sl** 中的智能绑定机制仅绑定显式引用其符号的库。

-d 检查对数据符号的引用。

-r 检查对数据和代码符号的引用。

-s 显示用于查找共享库的搜索路径。

-v 显示所有相关性关系。

外部语言环境影响

环境变量

ldd 使用下列环境变量查找共享库。

LD_LIBRARY_PATH

PA-RISC 64 位模式：在运行时定义共享库的搜索路径的路径名称的冒号分隔列表。有关详细信息，请参阅 *dld.sl*(5) 中的 PA-RISC 64 位动态路径列表。

LD_PRELOAD

动态加载程序在运行时首先隐式加载的库的冒号分隔或空格分隔列表。有关详细信息，请参阅 *dld.sl*(5) 中的 LD_PRELOAD 环境变量。

SHLIB_PATH

在运行时定义共享库的搜索路径的路径名称的冒号分隔列表。有关详细信息，请参阅 *dld.sl*(5) 中的 PA-RISC 32 位动态路径列表和 PA-RISC 64 位动态路径列表。

下列国际化变量会影响 **ldd** 的执行：

LANG 确定未提供 **LC_ALL** 和其他 **LC_*** 环境变量时，本国语言、本地惯例和编码字符集的语言环境类别。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值 **C** （请参阅 *lang*(5) ），而不是使用 **LANG**。

LC_ALL
用于确定所有语言环境类别的值，它优先于 **LANG** 和其他 **LC_*** 环境变量。

LC_MESSAGES
用于确定应该用来影响写入标准错误的诊断消息的格式和内容的语言环境。

LC_NUMERIC
用于确定数字格式的语言环境类别。

LC_CTYPE
用于确定字符处理函数的语言环境类别。

NLSPATH
用于确定消息清单的位置，以便处理 **LC_MESSAGES**。

如果任一国际化变量包含无效设置，则 **ldd** 就会认为所有国际化变量都设置为 **C**。请参阅 *environ*(5) 。

诊断信息

ldd 将共享库路径名的记录输出到 stdout。将可选的符号解析问题列表输出到 stderr。

ldd 在操作成功时返回零。非零返回代码表示出现了错误。

举例

缺省情况下， **ldd** 输出简单的动态路径信息。它由可执行文件（或共享库）中记录的相关性后接这些库所在的物理位置组成。

```
$ ldd a.out
  ./libx.sl =>    ./libx.sl
  libc.2 =>       /lib/pa20_64/libc.2
  libdl.1 =>      /lib/pa20_64/libdl.1
```

-v 选项可使 **ldd** 输出相关性关系以及动态路径信息。

```
$ ldd -v a.out
 find library=./libx.sl; required by a.out
  ./libx.sl =>    ./libx.sl
 find library=libc.2; required by a.out
  libc.2 =>       /lib/pa20_64/libc.2
 find library=libdl.1; required by /lib/pa20_64/libc.2
  libdl.1 =>      /lib/pa20_64/libdl.1
```

-r 选项可使 **ldd** 分析所有符号引用，并输出有关无效代码和数据符号的信息。

```
$ ldd -r a.out
```

```
  ./libx.sl =>    ./libx.sl
  libc.2 =>       /lib/pa20_64/libc.2
  libdl.1 =>      /lib/pa20_64/libdl.1
  symbol not found: val1 (./libx.sl)
  symbol not found: count (./libx.sl)
  symbol not found: func1 (./libx.sl)
```

-s 选项可使 **ldd** 输出用于按引用路径列表的顺序加载库的动态搜索路径列表:

```
$ export SHLIB_PATH=/tmp
$ export LD_LIBRARY_PATH=/var/tmp:/var/adm
$ ldd -s a.out
find library=./libx.sl; required by a.out
    ./libx.sl =>    ./libx.sl
find library=libc.2; required by a.out
  search path=/var/tmp:/var/adm  (LD_LIBRARY_PATH)
  trying path=/var/tmp/libc.2
  trying path=/var/adm/libc.2
  search path=/tmp  (SHLIB_PATH)
  trying path=/tmp/libc.2
  search path=/usr/lib/pa20_64:/opt/langtools/lib/pa20_64:  (RPATH)
  trying path=/usr/lib/pa20_64/libc.2
    libc.2 =>       /usr/lib/pa20_64/libc.2
find library=libdl.1; required by /usr/lib/pa20_64/libc.2
  search path=/var/tmp:/var/adm  (LD_LIBRARY_PATH)
  trying path=/var/tmp/libdl.1
  trying path=/var/adm/libdl.1
  search path=/tmp  (SHLIB_PATH )
  trying path=/tmp/libdl.1
  search path=/usr/lib/pa20_64  (RPATH)
  trying path=/usr/lib/pa20_64/libdl.1
    libdl.1 =>      /usr/lib/pa20_64/libdl.1
```

警告

ldd 不列出使用 *dlopen*(3C) 或 *shl_load*(3X) 显式加载的共享库。

文件

a.out	输出文件
/usr/lib/dld.sl	32 位 PA-RISC 动态加载程序
/usr/lib/pa20_64/dld.sl	64 位 PA-RISC 动态加载程序

/usr/ccs/lib/lddstub	为检查共享库相关性而加载的 32 位伪可执行文件。
/usr/ccs/lib/pa20_64/lddstub	为检查共享库相关性而加载的 64 位伪可执行文件。
/usr/lib/nls/$LANG/ldd.cat	消息清单

另请参阅

系统工具

ld(1) 调用链接编辑器

其他信息

a.out(4) 汇编程序、编译程序和链接程序输出

dld.sl(5) PA-RISC 动态加载程序

文本和教程

《HP-UX Linker and Libraries User’s Guide》

名称

ld_ia： ld - 链接编辑器

概要

链接编辑器。

ld [**-bdmnqrstvxzEGINOPQSTVZ**] [**-a** *search*] [**-c** *filename*] [**-dynamic**]
[**-e** *epsym*] [**-h** *symbol*]... [**-k** *filename*] [**-l***x* | *file*] ...
[**-l:** *library*] [**-m**] [**-noshared**] [**-noshared_dynamic**] [**-o** *outfile*]
[**-symbolic**] [**-u** *symbol*]... [**-y** *symbol*]... [**-A** *name*]
[**-B** *bind*]... [**-C** *n*] [**-D** *offset*] [**-Fl**] [**-Fw**] [**-Fz**] [**-G**] [**-L** *dir*]...
[**-N**] [**-O**] [**-Pd**] [**-PD** *file*] [**-PF** *file*] [**-Q**] [**-R** *offset*] [**-S**] [**-T**] [**+[no]allowunsats**]
[**+as** *mode*] [**+b** *path_list*] [**+cdp** *oldpath:newpath*] [**+cg** *path*] [**+compat**]
[**+copyobjdebug**] [**+[no]defaultrpath**] [**+df** *file*]
[**+dumpextern** *filename*] [**+dpv**] [**+e** *symbol*]... [**+ee** *symbol*]...
[**+fb**] [**+fbu**] [**+filter** *shared_library_path*]
[**+fini** *function*]... [**+[no]forceload**]
[**+gstbuckets** *size*] [**+gst**] [**+gstsize** *size*] [**+h** *internal_name*]
[**+help**] [**+hideallsymbols**] [**+ild**] [**+ildnowarn**] [**+ildpad** *percentage*]
[**+ildrelink**] [**+init** *function*]... [**+instrumenter** *filename*]
[**+interp** *filename*] [**+k**] [**+mergeseg**] [**+n**] [**+nocopyobjdebug**]
[**+nodynhash**] [**+nodefaultmap**] [**+noenvvar**]
[**+noobjdebug**] [**+nosectionmerge**] [**+nosmartbind**] [**+nosrcpos**]
[**+objdebugonly**] [**+origin** *shared_library_name*] [**+paddata** *pagesize*]
[**+padtext** *pagesize*] [**+pd** *size*] [**+pdzero**] [**+pgm** *name*] [**+pi** *size*]
[**+plabel_cache** *flag*] [**+profilebucketsize 16|32**] [**+rpathfirst**]
[**+s**] [**+std**] [**+stripunwind**] [**+tools**] [**+v[no]shlibunsats**]
[**+vallcompatwarnings**] [**+v[no]compatwarnings**]
[**+vtype** *type*] [**+FP** *flag*] [**+I** *symbol*]... [**+O[no]fastaccess**]
[**+O[no]procelim**] [**+Oreusedir=***dir*] [**+Oselectivepercent** *n*]
[**+Oselectivesize** *size*] [**+OselectiveO3**] [**+Ostaticprediction**]
[**+allowdups**] [**+interposer**] [**+[no]lazyload**]

说明

ld 将一个或多个对象文件或库作为输入，并将它们组合在一起生成单个文件（通常是可执行文件）。这样，它可以解析对外部符号的引用，将最终地址分配到过程和变量，修订代码和数据以反映新的地址（称作“重定位”的过程），并更新符号调试信息（当存在于文件中时）。缺省情况下， **ld** 将生成一个可执行文件，该文件可由 HP-UX 加载程序 **exec()** 运行（请参阅 *exec*(2) ）。或者，链接程序可以生成一个可重定位的文件，它适合由 **ld** 做进一步处理（请参阅下面的 **-r** ）。它也可生成一个共享库（请参阅下面的 **-b** ）。如果存在任何重复的符号或者仍然存在未解析的外部引用，链接程序会将输出文件标记为不可执行。如果在执行过程中出现其他任何错误， **ld** 可能会（也可能不会）生成输出文件（请参阅 **+k** 选项）。

ld 识别三种输入文件：由编译程序、汇编程序或链接程序生成的对象文件（也称作 **.o** 文件），由链接程序创建的共享库以及对象文件的归档（称作归档库）。归档库包含其组件对象文件中所有外部可见符号的表。（归档程序命令 *ar*(1) 创建并维护该索引）。 **ld** 使用该表来解析对外部符号的引用。

ld 按照文件在命令行上出现的相同顺序对其进行处理。当且仅当该对象模块在用户程序内提供当前未解析引用的定义时，它才会包括归档库元素中代码和数据（请参阅 **+[no]forceload** ）。通常的做法是在命令行上所有简单对象文件名后列出库。

共享库中的代码和数据决不会复制到可执行程序中。对于 32 位模式， **crt0.o** 位于 **/usr/ccs/lib/hpux32/crt0.o** 。对于 64 位模式， **crt0.o** 位于 **/usr/ccs/lib/hpux64/crt0.o** 。应该将 **crt0.o** 包括在 **-noshared** 链接中。对于 32 位模式，动态加载程序位于 **/usr/lib/hpux32/dld.so** 。对于 64 位模式，动态加载程序位于 **/usr/lib/hpux64/dld.so** 。动态加载程序将每个必需的库附加到进程，并解析程序及其库之间的所有符号引用。

共享库的文本段在使用该库的所有进程之间共享；使用该库的每个进程均会收到各自的数据段副本。如果 **pxdb -s on** 已经在加载库的可执行文件上运行，则将以私密的方式为运行该可执行文件的每个进程映射共享库的文本段。**ld** 以递归方式检查由 **ld** 创建的程序所用的共享库的相关性。如果 **ld** 没有在共享库相关性列表中记录的路径处找到支持的共享库，并且相关性是创建共享库时使用的 **-l** 参数的结果， **ld** 会搜索将为使用 **-l** 指定的库而搜索的所有目录（请参阅 **-L** 和 **LPATH** ）。

选项

-a *search* 指定是否使用 **-l** 选项搜索共享库或归档库。 *search* 的值应该是 **archive** 、 **shared** 、 **archive_shared** 、 **shared_archive** 或 **default** 之一。该选项可以出现多次，在 **-l** 选项之间散布，以控制每个库的搜索。缺省设置是使用库的共享版本（如果可用）或者归档版本（如果不可用）。

如果 **archive** 或 **shared** 处于活动状态，则仅接受指定的库类型。

如果 **archive_shared** 处于活动状态，则首选归档形式，但允许使用共享形式。

如果 **shared_archive** 处于活动状态，则首选共享形式，但允许使用归档形式。

要创建静态绑定的程序，请使用 **-noshared** 选项而不是 **-a archive** 选项。

-b 创建一个共享库而不是常规的可执行文件。使用该选项处理的对象文件必须包含编译程序在缺省情况下生成的 “位置无关代码” (PIC)。有关位置无关代码的信息，请参阅 *cc*(1) 、 *aCC*(1) 、 *f90*(1) 、 *as*(1) 和《Linker and Libraries Online User Guide》。

-c *filename* 从文件中读取 ld 选项。每一行包含零个或更多个由空格分隔的参数。文件中的每一行（包括最后一行）必须以换行符结尾。 **#** 字符表示该行的其余部分是注释。要将 **#** 字符转义，请使用序列 **##** 。

-d 强制“通用”符号的定义；即，为 **-r** 输出分配地址和大小。

-dynamic 该选项是缺省值。指示链接程序生成动态链接的可执行文件（可使用共享库的程序）。该选项是 **-noshared** 的补充。

如果没有链接任何共享库，则链接程序将构建动态链接的可执行文件。但是，在使用 **+compat** 选项的 PA32 位模式下，如果没有链接共享库，链接程序将构建静态绑定的可执行文件（或归档绑定的可执行文件）。

对于动态链接的可执行文件，将在加载可执行文件的过程中使用动态加载程序，而不管它是否与共享库链接。对于 **-noshared** （或静态绑定）程序，控制权不会传递给动态加载程序。有关详细信息，请参阅 *dld.so*(5) 。

-e *epsym* 将输出文件的缺省入口点地址设置为符号 *epsym* 的地址（该选项仅适用于可执行文件）。

-h *symbol* 在将符号表写入输出文件之前，将该名称标记为“本地”，使其不再在外部可见。这将确保将来由 **ld** 进行处理的过程中，该特定条目将不会与其他文件中的定义发生冲突。如果在构建共享库或程序时使用，该选项将防止命名符号对于动态加载程序可见。

可以使用多个选项符号对在命令行上指定多个 *symbol* ，也就是说，您指定的每个 *symbol* 前面必须带有 **-h** 选项。

-k *filename* 指定对输出文件内存映射进行描述的映射文件。

有关详细信息，请参阅《HP-UX Linker and Libraries User's Guide》指南和 **+nodefault-map** 。

-l*x* 搜索库 **lib***x***.a** 、 **lib***x***.so** 或 **lib***x***.sl** ，其中 *x* 是一个或多个字符。 **-a** 选项的当前状态确定是搜索库的归档 (**.a**) 版本还是共享 （**.sl** 或 **.so** ）版本。由于在遇到库名时对库进行搜索，因此 **-l** 的放置位置非常重要。缺省情况下，32 位库位于 **/usr/lib/hpux32** 。64 位库位于 **/usr/lib/hpux64** 。如果用户的环境中存在环境变量 **LPATH** ，它应该包含要搜索的目录的冒号分隔列表。将搜索这些目录（而不是缺省目录），但仍可以使用 **-L** 选项。如果程序使用共享库，动态加载程序 **/usr/lib/hpux32/dld.so** （32 位）或 **/usr/lib/hpux64/dld.so** （64 位）将尝试从链接时所在的相同目录中加载每个库（请参阅 **+s** 和 **+b** 选项）。

-l: *library* 搜索指定的库。与 **-l** 选项类似，但例外的是 **-a** 选项的当前状态不重要。库名可以是任意有效的文件名。

-m 在标准输出上生成加载映射。

-n 将忽略该选项。

-noshared 强制链接程序创建完全归档绑定程序（也称作静态绑定可执行文件）。使用该选项时，可在 ld 命令行上指定 **/usr/ccs/lib/hpux32/crt0.o** 或 **/usr/ccs/lib/hpux64/crt0.o** （或等效的启动代码）。该选项是 **-dynamic** 的补充。

对于动态链接的可执行文件，将在加载可执行文件的过程中使用动态加载程序，而不管它是否与共享库链接。对于静态链接程序，控制权不会传递给动态加载程序。

-noshared_dynamic
如果已链接共享库，则创建动态链接程序。如果未链接共享库，链接程序将创建完全归档绑定程序。该选项是兼容（与 **+compat** ）模式选项中的缺省值。另请参阅 **-dynamic** 和 **-noshared** 选项。

-o *outfile* 生成名为 *outfile* 的输出对象文件（如果未指定 **-o** *outfile* ，则为 a.out ）。

-q 将忽略该选项。

-r 为后继的重新链接将重定位信息保留在输出文件中。 **ld** 命令不报告未定义的符号。该选项不能在构建共享库（ **-b** ）时使用，也不能与 **-s** 、 **-x** 或 **+ild** 递增链接选项一起使用。

-s 从输出文件中删除所有符号表、重定位和调试支持信息。（ *strip*(1) 命令也会删除这些信息）。该选项与 **-r** 选项和 **+ild** 选项不兼容。

注意：使用 **-s** 选项可能会妨碍或阻止对得到的程序使用符号调试程序。

-symbolic *symbol*
构建共享库时，将导致链接程序将对指定符号的所有引用解析为库中定义的符号。该选项类似于 **-B symbolic** ，但只能对每个符号运行。

可以使用多个选项符号对在命令行上指定多个 *symbol* ，也就是说，您指定的每个 *symbol* 前面必须带有 **-symbolic** 选项。

-t 在 **ld** 进行处理时输出每个输入文件的踪迹（到标准的输出）。

-u *symbol* 输入 *symbol* 作为符号表中的未定义符号。得到的未解析引用对于仅从库中的对象文件链接程序非常有用。

可以使用多个选项符号对在命令行上指定多个 *symbol* ，也就是说，您指定的每个 *symbol* 前面必须带有 **-u** 选项。

-v 在链接过程中显示详细消息。该选项与 **+vtype all** 等效（有关详细信息，请参阅 **+vtype** 选项）。

-x 从输出文件中删除本地符号。这将减小输出文件的大小，而不会损害对象文件工具的有效性。该选项与 **-r** 选项和 **+ild** 选项不兼容（递增链接程序需要用 **-x** 选项删除的输出加载模块部分）。

注意：使用 **-x** 选项可能会妨碍或阻止对得到的程序使用符号调试程序。

-y *symbol* 指示出现 *symbol* 的每个文件。可以使用多个选项符号对在命令行上指定多个 *symbol* ，也就是说，您指定的每个 *symbol* 前面必须带有 **-y** 选项。

-z 安排运行时取消引用的空指针来生成 **SIGSEGV** 信号（它是 **-Z** 选项的补充。 **-Z** 是缺省值）。

-A *name* 该选项将被忽略，并生成警告消息。

-B *bind* 选择使用共享库的程序的运行时绑定行为或构建共享库时的绑定首选项。 *bind* 最常用的值包括：

direct 通过在符号解析过程中记录解析共享库的名称，创建符号引用和共享库之间的直接链接。该信息在运行时用于快速解析符号，而不必搜索当前加载的所有库。

-B direct 将隐式打开符号绑定（请参阅 **-B symbolic** ）并禁用相关的共享库处理。

通过设置 **LD_NODIRECTBIND** 环境变量，可以在运行时禁用直接绑定。

deferred
在第一个引用上（而不是在程序启动时）绑定地址。这是缺省值。

group 标记共享库，使其如同加载 **RTLD_GROUP** 标志以 **dlopen()** 一样进行操作。这不影响相关共享库。

immediate
在加载库时立即绑定所有符号的地址。通常后接 **-B nonfatal** ，以便在第一个引用上解析在程序启动时无法解析的过程调用。

由于 **-B nonfatal** 禁止有关未解析符号的消息，也可以指定 **-B verbose** 显示这些消息。

请参阅下面的示例。

lazydirect
仅记录带有缓慢加载标记的共享库的直接绑定信息。请参阅 **+[no]lazyload**。

nodelete
标记共享库，使得使用 **dlclose()** 或 **shl_load()** 的显式卸载静默地返回成功，而不将共享库与进程分离。因此，共享库句柄仅对于 **shl_findsym()** 有效。在使用 **shl_load()** 或 **dlopen()** 进行下一次显式加载之前，它对于 **dlsym()** 、 **dlclose()** 和 **shl_unload()** 一直无效。

nodirect
禁用直接绑定。仅记录“直接提示”，以引用带有缓慢加载标记的库。这是缺省行为。

nonfatal
如果还将 **-B immediate** 用于无法在程序启动时绑定的代码符号，请延迟绑定，直到引用它们。请参阅上面的 **-B immediate** 说明。

由于 **-B nonfatal** 禁止有关未解析符号的消息，也可以指定 **-B verbose** 显示这些消息。

restricted
使得符号定义搜索范围仅限于在库加载时可见的符号。

symbolic
仅在构建共享库时使用。该选项在可能的情况下导致共享库中的所有引用在内部解析。这些内部解析的符号仍会在外部可见。缺省情况下，（无 **-B symbolic** 选项），对共享库中符号的引用将解析为最为可见的定义。导出该符号的第一个加载模块（a.out 或共享库）包含最为可见的定义。多个加载模块可定义和导出同一个符号。即使共享库中的符号在共享库中定义，对该符号的引用仍可以解析为其他共享库中的定义。可以使用该选项强制共享库中的所有引用使用其自己的定义（如果在共享库中定义）。有关将 **-B symbolic** 与这些选项一起使用的详细信息，请参阅 **+e** 和 **+ee** 选项。

verbose
在绑定符号时显示详细消息。这是缺省设置，但指定 **-B nonfatal** 时除外。这种情况下，必须显式指定 **-B verbose** 来获取详细消息。

有关如何使用绑定模式的详细信息，请使用 **+help** 选项或参阅《HP-UX Linker and Libraries User’s Guide》手册。

-C *n* 该选项将被忽略，并生成警告消息。

-D *offset* 以十六进制设置数据段的起始地址。该选项可用于内核和嵌入式应用程序。 64 位模式的缺省地址为 0x6000000000000000， 32 位模式的缺省地址为 0x40000000。

-E 标记程序定义要导出至共享库的所有符号。在 **+compat** 模式链接中， **ld** 仅标记链接时所见的共享库实际引用的符号。在 **+std** 链接中，缺省情况下将导出所有符号，因此 **-E** 不一定使符号可见。但是，它有一种附加的副作用，即根据需要标识所有导出的符号，这样在使用无用代码删除 (**+Oprocelim**) 时将不会删除这些符号。

-Fl 强制加载归档库。等效于 **+forceload** 。

-Fw 该选项将被忽略，并生成警告。

-Fz 将接受并忽略该选项。

-G 从输出文件中删除所有未加载的数据。该选项通常用于删除调试信息，并且与 **+ild** 选项不兼容。

注意：使用 **-G** 选项可能会妨碍或阻止对得到的程序使用符号调试程序。

-I 提供用于在执行时收集配置文件信息的代码。在执行提供的程序时，配置文件数据库文件是输出（缺省情况下名为 flow.data）。在程序执行过程中收集的配置文件数据可以与 **-P** 选项一起使用。缺省的提供程序是动态提供程序 **/opt/langtools/bin/caliper** ，但可以

使用 **+instrumenter** 选项调用静态提供程序 **/opt/langtools/bin/sin** 。

该选项不应与 **-P** 、 **-O** 、 **+ild** 或 **+O** 选项一起使用。

NOTE: 如果使用 **+instrumenter sin** ，提供程序的建议方法是使用编译程序的 **+I** 选项，而不是 **ld -I** 选项。如果直接调用链接程序，则必须将 **-u__sin_core__** 、 **-u__sin_init** 和 **-lsin** 选项传递给链接程序。如果既有提供的共享库，又有提供的要与该库链接的共享可执行文件，则除了 **-u** 选项之外，还必须包括 **-h__sin_core__** 和 **-h__sin_lookup_ibt** 选项。如果使用缺省值或 **+instrumenter caliper** ，将不需要任何附加的链接程序选项。

-L *dir* 在缺省位置中查找之前在 *dir* 中搜索 **lib***x***.a** 、 **lib***x***.sl** 或 **lib***x***.so** 。可以指定多个目录，但是每个目录前必须添加 **-L** 选项。 **-L** 选项只有在位于命令行上的 **-l** 选项之前时才有效。

-N 仅在 32 位模式下，使得数据直接放置在文本之后，并使文本可写。这种文件不能共享。

-O 打开链接程序优化。该优化目前包括删除无用过程。

当选择 **+O4** 编译程序选项时， **-O** 将由编译程序传递给链接程序。

该选项与 **+ild** 选项不兼容。

有关链接程序优化的详细信息，请使用 **+help** 选项或参阅《HP-UX Linker and Libraries User's Guide》手册。

-P 检查提供的程序所生成的配置文件数据库文件（请参阅 **-I** 选项），以执行基于配置文件的代码优化。该选项不应与 **+ild** 选项一起使用。

-Pd 将可调试函数重新排序。通常，由于重新排序会使包含调试信息的 **.o** 文件中的函数无法调试，因此 **-P** 不会将其重新排序。该选项将覆盖这一行为，将这些函数重新排序。重新排序在缺省情况下基于从 flow.data 生成的链接顺序文件。如果指定 **-Pd** 选项，链接程序将不使用 flow.data 来进行重新排序。该选项与 **+ild** 不兼容。

注意：使用 **-Pd** 选项可能会妨碍或阻止对得到的程序使用符号调试程序。

-PD *filename* 将 **fdp** 在链接过程中使用 **-P** 选项生成的链接顺序文件保存到用户指定的文件。该选项与 **+ild** 选项不兼容。

-PF *filename* 指示链接程序将指定文件用于链接顺序文件，而不是使用 **/usr/ccs/bin/fdp** 生成该文件。该选项与 **+ild** 选项不兼容。

-Q 将忽略该选项。

-R *offset* 以十六进制设置数据段（即代码段）的起始地址。该选项可用于内核和嵌入式应用程序。 64 位模式的缺省地址是 0x4000000000000000 ， 32 位模式的缺省地址是 0x04000000。如果指定了 **-N** 选项，缺省值为 0x1000。

-S 该选项将被忽略，并生成警告消息。

-T 将忽略该选项。

-V 输出消息，提供有关所用 **ld** 版本的信息。

-Z 这是缺省值。允许空指针的运行时取消引用。请参阅 *cc*(1) 中有关 **-Z** 和 *pointers* 的讨论（它是 **-z** 选项的补充）。

+allowdups 允许多个符号定义。缺省情况下，在可重定位的对象之间出现的多个符号定义可能会导致致命的错误状态。该选项禁止该错误状态，并允许采用第一个符号定义。

+[no]allowunsats
控制不满意符号错误报告。如果得到的输出文件包含不满意的符号， **+allowunsats** 将不标志错误。这是可重定位的链接和共享库版本的缺省值。如果得到的输出文件包含不满意的符号， **+noallowunsats** 将标志错误。这是程序文件的缺省值。

+as *mode* 控制将由内核使用的地址空间模式。可能的模式值为 **default** 、 **share_magic** 、 **exec_magic** 、 **shmem_magic** 和 **mpas** 。缺省值当前等效于 **share_magic** 。要将模式设置为缺省值之外的任意值，必须将该选项与 **-N** 选项一起使用，确保文本段和数据段是相邻的。

+b *path_list* 指定要在程序运行时搜索的目录的冒号分隔列表，可定位使用 **-l** 或 **-l:** 选项指定的可执行输出文件所需的共享库。该目录列表将成为嵌入的路径。如果未使用 **+b** ，或者已指定单个冒号 **(:)** 作为参数， **ld** 将使用由 **-L** 选项和 **LPATH** 环境变量指定的所有目录构建嵌入的路径（请参阅 **+s** 选项）。

+cdp *oldpath:newpath*
将忽略该选项。

+compat 打开链接程序中的兼容性模式 — PA-RISC 32 位链接的模拟行为。

+[no]copyobjdebug
当使用 **+noobjdebug** 链接程序选项覆盖 **+objdebug** 编译程序选项的效果时，链接程序将省略对象文件中的 **+objdebug** 信息（除了将调试信息复制到输出文件之外）。但是，如果任何对象文件是先前 **-r** 链接的结果，则不省略这些文件中的 **+objdebug** 信息。 **+nocopyobjdebug** 选项与 **+noobjdebug** 选项一起使用时将强制链接程序省略所有对象文件中的 **+objdebug** 信息，包括使用 **-r** 选项生成的对象。 **+copyobjdebug** 是缺省值。

+[no]defaultrpath
+defaultrpath 是缺省值。在嵌入路径中包括使用 **-L** 指定的任何路径，除非您指定 **+b** 选项。如果使用 **+b** ，嵌入路径中仅包括由 **+b** 指定的路径列表。

+nodefaultrpath 选项从嵌入路径中删除用 **-L** 选项指定的所有库路径。链接程序将在链接时搜索由 **-L** 选项指定的库路径。在运行时，搜索的库路径只是由环境变量 **LD_LIBRARY_PATH** 和 **SHLIB_PATH** 指定的库路径、由 **+b** 链接程序选项指定的库路

径以及缺省的库路径。

+df *file* 与 **-P** 选项一起使用时，该选项指定 *file* 应该用作配置文件数据库文件。缺省值为 **flow.data** 。有关详细信息，请参阅有关 **FLOW_DATA** 环境变量的讨论。该选项与 **+ild** 选项不兼容。

+dpv 有关由 procelim 删除的过程的输出信息。等效于 **+vtype procelim** 。

+dumpextern *filename*
对于可执行文件和共享库链接有效。指示链接程序将所有外部符号转储到由 *filename* 指定的文件。这会将在加载模块（a.out 或共享库）中引用（但不在加载模块中定义）的所有外部符号转储到指定的文件。可以使用 **-Bextern:** *filename* 选项将该文件传递回您的编译程序。有关详细信息，请参阅编译程序选项 **-Bextern:** *filename* 、 **-Bhidden** 和 **-Bprotected** 。

+e *symbol* 在构建共享库或程序时，标记符号以导出至动态加载程序。仅导出显式标记的符号。构建共享库时，将在内部解析未导出符号的调用。

如果将 **+e** 或 **+ee** 选项与 **-B symbolic** 一起使用，则将在内部解析对指定符号的引用（如果已定义）。运行时行为可能与单独使用 **+e** 时不同。

可以使用多个选项符号对在命令行上指定多个 *symbol* ，也就是说，您指定的每个 *symbol* 前面必须带有 **+e** 选项。

+ee *symbol* 因为该选项可以导出符号，所以与 **+e** 选项类似。但是，与 **+e** 选项不同， **+ee** 选项不更改文件中其他任何符号的可见性。构建 **+compat** 模式可执行文件时， **ld** 在缺省情况下仅导出那些由链接时可见的共享库所实际引用的符号。 **+ee** 选项在使用 **+compat** 指定时，具有导出指定符号而不隐藏在缺省情况下导出的任何符号的效果。在 **+std** 模式链接中，在缺省情况下将导出所有符号，因此 +ee 不一定会使符号可见。但是，它有一种附加的副作用，即根据需要标识符号，这样在使用无用代码删除 (**+Oprocelim**) 时将不会删除该符号。如果还给定了 **+hideallsymbols** 等选项， **+ee** 选项仍会保留其导出行为。

可以使用多个选项符号对在命令行上指定多个 *symbol* ，也就是说，您指定的每个 *symbol* 前面必须带有 **+ee** 选项。

+fb 指示链接程序对它生成的可执行文件运行 fastbind 工具。该可执行文件应该与共享库链接。有关 *fastbind*(1) 的详细信息，请使用 **+help** 选项或参阅《HP-UX Linker and Libraries User’s Guide》手册。该选项与 **+ild** 选项不兼容。

+fbu 将 **-u** 选项传递给 fastbind 工具。有关 *fastbind*(1) 的详细信息，请使用 **+help** 选项或参阅《HP-UX Linker and Libraries User’s Guide》手册。该选项与 **+ild** 选项不兼容。

+filter *shared_library_path*
启用共享库过滤器机制，它可用于将较大的库拆分为一个“过滤器”和多个“实现”库，以便更有效地组织共享库。 *shared_library_path* 指定过滤器库的位置。有关详细信息，请参阅《HP-UX Linker and Libraries User’s Guide》。

+fini *function_name*

指定将按照前进顺序（函数在命令行上从左到右的顺序）调用的终结程序函数。

可以使用多个选项符号对在命令行上指定多个终结程序函数，也就是说，您指定的每个函数前必须添加 **+fini** 选项。

+[no]forceload 缺省值是 **+noforceload** 。 **+forceload** 选项从归档库中加载所有对象文件。 **+noforceload** 仅从归档库中加载必需的对象文件。在显式更改之前，选定的模式（显式或缺省）将一直有效。

+gst 启用全局符号表散列机制，用于查找导入（或导出）条目的值。 **+gst** 和相关选项通过使用全局符号表（提高导出符号搜索的性能）提供了性能改进。有关详细信息，请参阅 *dld.so*(5) 和《HP-UX Linker and Libraries Online User Guide》。

+gstbuckets *size*

将忽略该选项。

+gstsize *size* 使用全局符号表散列机制请求特定的散列阵列大小。缺省值为 1103 。通过将 **_HP_DLDOPTS** 环境变量设置为值 **-symtab_size** 质数，可以在运行时覆盖该值。可以使用 **chatr +gstsize** *size file* 设置该值。

+h *internal_name*

在构建共享库时，记录 *internal_name* 作为该库的名称。当库用于链接其他可执行文件（程序或共享库）时，该 *internal_name* 将记录在结果输出文件的库列表中，而不是输入共享库的路径名中。也就是说，如果未使用 **+h** ，共享库将没有内部名称，并且当使用共享库构建可执行文件时，链接程序将记录它所查看的库名。

如果在链接行上看到多个 +h 选项，链接程序将使用第一个并发出警告消息。

+help 启动帮助浏览器工具《HP-UX Linker and Libraries Online User Guide》，它随 HP-UX 操作系统附带。有关详细信息，请参阅《HP-UX Linker and Libraries User's Guide》手册。有关订购信息，请参阅 *manuals*(5) 。

+hideallsymbols 防止所有符号的导出，除非使用 **+e** 显式导出。该选项在符号表中将所有符号标记为“本地”。另请参阅 **-h** 和 **+e** 选项。

+ild 指定递增链接。

如果不存在输出文件，或者输出文件不是使用 **+ild** 选项创建的，链接程序将执行初始递增链接。所生成的输出文件适用于后继的递增链接。递增链接选项对于可执行文件和共享库链接均有效。

以下选项与 **+ild** 选项不兼容。如果指定以下与 **+ild** 不兼容的 **ld** 选项之一，链接程序将发出警告消息并忽略 **+ild** 选项。

-r 创建可重定位的对象文件。

Strip options:

-s 、 **-x** 和 **-G** 删除输出文件的内容。

Optimization options:

-I 、 **-O** 、 **-P** 、 **-PD** 、 **-PF** 、 **+df** *file* 、 **+fb** 、 **+fbu** 、 **+fbs** 、 **+pgm** *name* 、 **+Oprocelim**

以下选项在一定限制下与 **+ild** 选项兼容:

-D offset, **-R offset**

设置数据段和文本段的原点。如果在初始递增链接后更改偏移，链接程序将自动执行初始递增链接。

-k *mapfile*

提供非缺省的映射文件。用户指定的映射文件规范可以使用 **+ild** 选项来启用。但不应该在初始递增链接后修改映射文件。如果在初始链接后修改映射文件，则将自动修改初始递增链接。

+ildnowarn 禁止递增链接相关警告。缺省情况下，链接程序将发出所有递增链接相关警告。如果在没有 **+ild** 或 **+ildrelink** 的情况下使用该选项，则会将其忽略。

+ildpad *percentage*

控制递增链接程序相对于填充的对象文件结构的大小所分配的填充 *percentage* 。缺省情况下，链接程序将分配少于填充空间的 20%。如果在没有 **+ild** 或 **+ildrelink** 的情况下使用该选项，则会将其忽略。

+ildrelink 执行初始递增链接，而不管输出加载模块。

在某些情况（如内部填充空间已用尽）下，将强制递增链接程序执行初始递增链接。**+ildrelink** 选项可用于通过定期重建输出文件来避免这些意外的初始递增链接。

+init *function_name*

指定将按照相反顺序（函数在命令行上从右到左的顺序）调用的初始设置函数。

可以使用多个选项符号对在命令行上指定多个初始设置函数，也就是说，您指定的每个初始设置函数前必须添加 **+init** 选项。

+instrumenter *name*

指定要使用的提供程序。仅识别 **sin** 或 **caliper** 。缺省值是 **caliper** 。如果指定了 **sin** ，链接程序将自动调用 **/opt/langtools/bin/sin** 。如果指定了 **caliper** ，动态加载程序将在程序执行时自动调用 **/opt/langtools/bin/caliper** 。

+interp *filename*

更改 **dld** 路径，将由 *filename* 指定的程序用于“解释程序”作为动态加载程序。这在使用特殊版本的 dld.so 进行调试时非常有用。缺省路径是 **/usr/lib/hpux32/uld.so:/usr/lib/hpux32/dld.so** （32 位程序）或

/usr/lib/hpux64/uld.so:/usr/lib/hpux64/dld.so （64 位程序）。

+interposer 仅在构建共享库时使用。它将创建一个可用于内插的共享库。当使用直接绑定信息解析应用程序引用（请参阅 **-B direct** ）时，动态加载程序将首先搜索内插库。如果符号无法解析为任何内插库，则将使用直接绑定信息。

+k 如果在链接中没有遇到错误，则指示链接程序仅创建可执行文件。如果发现错误（系统错误或未解析的引用），则将删除输出文件。

+[no]lazyload 启用 [禁用] 共享库的慢速加载。对于 **+lazyload** 库，加载将延迟，直到在执行时引用该库。 **+lazyload** 和 **+nolazyload** 选项可能会同时在链接行上出现。在再次出现这两个选项中的一个之前，显式或缺省指定的模式将一直对链接行上的所有后继库有效。

满足以下一项或多项条件的库不具备慢速加载的资格:

• is a filter library

• is accessed via a data reference from another
模块

• is accessed via an indirect function call

链接程序将以静默方式将这些库转换为 **+nolazyload** 库。

将不会在链接时处理 **+lazyload** 共享库的相关库，除非在链接行上显式指定这些库。

通过设置 **LD_NOLAZYLOAD** 环境变量，可以在运行时禁用慢速加载。

+mergeseg 在可执行文件中设置一个标志，使得动态加载程序将启动时加载的共享库的所有数据段合并到一个数据块中。每个动态加载库的数据段也将与相关库的数据段合并。这样，内核可以使用具有较大页大小的表条目，从而将提高运行时性能。

+n 使得链接程序在搜索任何归档或共享库之前加载所有对象模块。然后，链接程序将按从左到右的顺序搜索在命令行上指定的归档和共享库。它将在命令行上从左到右重复搜索库，直到再也没有不满意的符号，或者最后的搜索没有添加新的定义。如果指定了两个具有符号相关性的库，该选项将非常有用。

+nodefaultmap 不用使用缺省内存映射。必须通过 **-k** 链接程序选项提供一个映射文件。

+nodynhash 禁用 **+gst** 选项的缺省链接程序行为，以创建可执行文件或共享库的 .dynhash 部分。使用该选项可禁止为以下库或可执行文件生成预计算的散列表信息：很少通过全局符号表查找方案使用，或者存储预计算散列值的系统开销太大的库或可执行文件。与 **-r** 选项一起使用时，该选项不起作用。

+noenvvar 指示动态加载程序在运行时忽略动态路径搜索环境变量 **LD_LIBRARY_PATH** 、 **SHLIB_PATH** 和 **$ORIGIN** 。缺省情况下，如果指定 **+std** 选项，动态加载程序将查看环境变量，即启用环境变量。如果指定 **+compat** 选项或 **+noenvvar** 选项，该选项将生效，动态加载程序将忽略环境变量（禁用环境变量）。请参阅 **+compat** 或 **+std** 选项。

通过 **chatr** 命令的“共享库动态路径搜索”输出，可以在可执行文件或共享库中显示该选项的状态。有关详细信息，请参阅 *chatr*(1) 。该选项通常用于安全程序。

+noobjdebug 覆盖 **+objdebug** 编译程序选项，并将所有调试信息复制到可执行文件。将 **+objdebug** 编译程序选项与任意 **-g** 选项一起使用时，链接程序会将调试信息保留在对象文件中，而不是将其复制到输出文件。可以在链接时使用 **+noobjdebug** 选项强制链接程序将调试信息复制到输出文件，从而抵消 **+objdebug** 编译程序选项的作用。另请参阅 **+nocopyobjdebug** 。

+nosectionmerge
使用 **-r** 选项可允许过程独立定位。缺省设置是将所有过程合并到单个位置。

+nosmartbind 将忽略该选项。

+nosrcpos 在 IC64（32 位和 64 位），编译程序选项 **+srcpos** 是缺省值。 **+srcpos** 使得编译程序即使在未指定编译程序选项 **-g** 时仍生成部分调试信息。缺省的 **+srcpos** 选项也始终会将部分调试信息复制到可执行文件，从而导致较大的可执行文件。通过 **+srcpos** ，用户可以使用 **cxperf** 、 **caliper** 和 **sin** 等工具配置程序（即使不存在 **-g** 编译）。

链接程序选项 **+nosrcpos** 可用于覆盖缺省的 **+srcpos** 编译程序选项，以及在链接时删除这些调试信息。 **+nosrcpos** 也可与 **"-g +objdebug"** 一起使用来完全强制 **+objdebug** 模式（即，将调试信息保留在对象文件中）。

+objdebugonly 忽略非 objdebug 对象或归档中的调试信息，并继续在 **+objdebug** 模式下操作。如果仅调试使用 **+objdebug** 选项编译的文件， **+objdebugonly** 可以指示链接程序绕过处理用 **+noobjdebug** 编译的文件中的调试信息的步骤。

+origin *shared_library_name* **-l***x*
（仅在 **-l** 选项或共享库名之前使用）。使得链接程序将 **$ORIGIN** 添加在共享库列表中 *shared library name* 的前面，并设置输出模块的 **DF_ORIGIN** 标志。在运行时，动态加载程序将确定父模块（对象模块、共享库或可执行文件）的当前目录，然后将 **$ORIGIN** 替换为该目录的名称。例如，

$ **ld -dynamic main.o +origin libx.so -L /usr/lib/hpux32/ -lc**

当 **+origin** 选项可用时，建议使用 **+b** 选项在嵌入路径中指定 **$ORIGIN** ，例如：

$ **ld -dynamic main.o -L /usr/lib/hpux32/ -lc +b \$ORIGIN**

有关 **$ORIGIN** 的详细信息，请使用 **+help** 选项或参阅《HP-UX Linker and Libraries User’s Guide》手册。

+paddata *pagesize*
用零将数据段填充到 *pagesize* 的倍数。这可以改进页分配，允许内核分配较少的大数据页，从而提高 TLB 命中率。使用该选项将增加输出文件大小。

+padtext *pagesize*
用零将文本段填充到 *pagesize* 的倍数。这可以改进页分配，允许内核分配较少的大数据页，从而提高 TLB 命中率。使用该选项将增加输出文件大小。

+pd *size* 请求应该用于数据的特定虚拟内存页大小。支持的大小包括 **4K** 、 **16K** 、 **64K** 、 **256K** 、 **1M** 、 **4M** 、 **16M** 、 **64M** 、 **256M** 、 **D** 和 **L** 。大小 **D** 允许内核选择应该使用的页大小。大小 **L** 会使用可用的最大页大小。如果无法满足请求的大小，则实际的页大小可能有所不同。

+pdzero 将忽略该选项。

+pgm *name* 使用 **-P** 选项，指定 *name* 应该用作配置文件数据库文件中的查找名称。缺省值是输出文件的基名（由 **-o** 选项指定）。该选项与 **+ild** 选项不兼容。

+pi *size* 请求应该用于指令的特定虚拟内存页大小。有关其他信息，请参阅 **+pd** 选项。

+plabel_cache *flag*
将忽略该选项。

+profilebucketsize [**16**|**32**]
指定配置处理示例计数器容器的大小。有效值是 16 或 32。有关详细信息，请参阅 *gprof*(1) 。

+s 该选项是缺省值。指示在运行时，动态加载程序可以使用环境变量 **SHLIB_PATH** 和 **LD_LIBRARY_PATH** 定位可执行文件输出文件所需的共享库，这些共享库允许动态库搜索。允许动态库搜索的共享库包含不带“/”（斜线）的内部名称（如共享库路径名的基名），或者不包含内部名称，并且是由 **-l** 或 **-L** （或者仅使用共享库路径名的基名）指定的。环境变量应该设置为目录的冒号分隔列表。

在兼容模式下使用 **+compat** 选项时，如果同时使用 **+s** 和 **+b** ，它们在命令行上的相对顺序指示将首先搜索哪个路径列表（请参阅 **+b** 选项）。在标准模式（缺省模式或使用 **+std** 选项）下， **+s** 和 **+b** 的顺序不影响动态加载程序搜索顺序，始终将首先搜索环境变量。

+rpathfirst 该选项将导致在搜索共享库时， **RPATH** （嵌入路径）在 **LD_LIBRARY_PATH** 或 **SHLIB_PATH** 中指定的路径之前使用。这将更改 **LD_LIBRARY_PATH** 、 **SHLIB_PATH** 和 **RPATH** （嵌入路径）的缺省搜索顺序。

+std 该选项是缺省值。打开链接程序的标准模式。该选项是 **+compat** 选项的补充。使用该选项启用的选项包括： **-dynamic** 。指定该选项时禁用或忽略的选项包括： **+compat** 、 **+noenvvar** 、 **-noshared** 。

+stripunwind 不要输出 unwind 表。这将创建较小的可执行文件大小。如果不需要使用 unwind 表进行调试或 aC++ 异常处理，请使用该选项。

注意：使用 **+stripunwind** 选项可能会妨碍或阻止对得到的程序使用符号调试程序。

+tools 将忽略该选项。

+vallcompatwarnings
将忽略该选项。

+v[no]compatwarnings
将忽略该选项。

+v[no]shlibunsats
启用 [禁用] 打印共享库所用的不满意符号的列表。缺省值是 **+vnoshlibunsats** 。由于不使用引用符号的模块，因此链接程序所报告的某些不满意符号在运行时是不必要的。

+vtype *type* 生成有关链接操作的详细输出。 *type* 可以具有以下值：

all 转储 **+vtype** 操作中的所有信息。与 **-v** 相同。

files 转储有关所加载的每个对象文件的信息。

heap 转储有关链接所用堆大小的信息。

libraries
转储有关所搜索的库的信息。

procelim
转储有关已经由 **+Oprocelim** 操作删除的部分的信息

sections
转储有关添加到输出文件的每个输入部分的信息。

symbols
转储有关在输入文件中引用（或定义）的全局符号的信息。

+FP *flag* 指定在程序启动时应该如何初始化浮点运算的环境。缺省情况下将禁用所有行为。支持以下标志（启用大写标志；禁用小写标志）：

D（**d**） 启用去标准化值的突然下溢（下溢到零）。

I（**i**） 限制产生不精确结果的浮点运算。

N（**n**） 限制 Denormal|Unnormal 操作数浮点运算。

O（**o**） 限制浮点上溢。

U（**u**） 限制浮点下溢。

V（**v**） 限制无效的浮点运算。

Z（**z**） 限制除以零。

要在运行时动态更改这些设置，请参阅 *fesettrapenable*(3M) 。

+I *function* 指定构建共享库时初始设置函数的名称。一个共享库可以指定多个初始设置。初始设置可以按照它们在命令行上的指定顺序来执行。

可以使用多个选项符号对在命令行上指定多个初始设置函数，也就是说，您指定的每个初始设置前必须添加 **+I** 选项。

该选项支持兼容性。建议使用 **+init** 和 **+fini** 选项。有关初始设置函数的详细信息，请使用 **+help** 选项或参阅《HP-UX Linker and Libraries User's Guide》手册。

+O[no]fastaccess

将忽略该选项。

+O[no]procelim 启用 [禁用] 删除应用程序不引用的过程。缺省值是 **+Onoprocelim** 。过程删除可能会在任意优化级别（包括级别 0）上发生。有关详细信息，请使用 **+help** 选项或参阅《HP-UX Linker and Libraries User's Guide》手册。该选项与 **+ild** 选项不兼容。

+Oreusedir=*dir* 该选项将被忽略，并生成警告消息。

+Oselectivepercent *n*

指示过程内优化程序驱动程序将对象文件的前百分之 *n* 传递到高级别的优化程序进行过程内优化（如内联）。该选项设计为在存在动态配置处理的情况下在优化级别 4 (+O4) 工作。

+Oselectivesize *size*

将接受并忽略该选项。

+OselectiveO3 将接受并忽略该选项。

+Ostaticprediction

该选项将被忽略，并生成警告消息。

缺省值

除非另行指定， **ld** 会将其输出文件命名为 **a.out** 。 **-o** 选项可覆盖该缺省值。缺省设置是创建动态链接的程序，除非您指定 **-noshared** 选项。 **-a** 的缺省状态是搜索共享库（如果可用）或归档库（如果共享库不可用）。缺省的绑定行为是 **deferred** 。

-Z/-z 选项的缺省值是 **-Z** 。

+objdebug 编译程序选项

+objdebug 编译程序选项在与任意 **-g** 选项使用时，将使调试信息保留在对象文件中，而不是将其放入输出文件。这将导致较短的链接时间和较小的输出文件。

要调试使用 **+objdebug** 选项编译的加载模块，HP WDB 调试程序必须具有对对象文件的访问权。（请注意，对于使用 **-r** 选择构建的对象文件，单个的对象文件必须可用于调试程序）。如果移动对象文件，请使用 HP WDB 的 **objdir** 命令指定这些对象的位置。

+noobjdebug 编译程序选项会将调试信息复制到输出文件。 **+objdebug** 是编译时缺省值。

如果链接程序检测到任何用 **+objdebug** 选项编译的对象文件，则会将调试信息保留在这些文件中。对于未用 **+objdebug** 编译的所有对象文件，均会将其调试信息复制到输出文件。

可以在链接时使用 **+noobjdebug** 选项继续将调试信息放入输出文件（即使某些对象已用 **+objdebug** 进行编译）。

递增链接

在编辑-编译-链接-调试开发周期中，链接时间是一种重要的组成部分。递增链接程序（可通过 **+ild** 和 **+ildrelink** 选项使用）利用了可重复使用程序先前版本的大部分内容并且不需要处理未更改的对象文件等事实，可以减少链接时间。通过递增链接程序，可以将对象代码插入先前创建的输出文件（可执行文件或共享库）中，而不必重新链接未修改的对象文件。在初始递增链接后进行重新链接所需的时间取决于所修改的模块的数量。

链接程序具有以下不同的链接模式：

- 常规链接：链接程序链接所有模块的缺省操作模式。
- 初始递增链接：当您请求递增链接但不存在递增链接程序创建的输出模块（或者该模块存在，但递增链接程序无法执行递增更新）时所进入的模式。
- 递增链接：当您请求递增链接，存在递增链接程序创建的输出模块，并且递增链接程序不需要初始递增链接时所进入的模式。

递增链接通常比常规链接快得多。在初始链接上，递增链接程序所需的时间与常规链接进程所需的时间大致相同，但是后继递增链接可以比常规链接快得多。在大小中等的链接（数十个文件，总大小为几个 MB）中，一个对象文件中的更改通常比常规 **ld** 链接快 10 倍。在分配的填充空间及其他约束条件许可的情况下，递增链接程序会执行尽可能多的递增链接。链接时间降低的代价是可执行文件或共享库大小的增加。

递增链接程序为输出文件的所有组件分配填充空间。填充使得由 **ld** 链接的模块变得更大。随着对象文件的大小在连续递增链接过程中不断增加，递增链接程序可能会耗尽可用的填充空间。如果出现这种情况，将显示警告消息，并执行该模块的完整初始递增链接。

在初始递增链接上，链接程序按照常规链接的相同方式处理输入对象文件和库。除了常规链接进程之外，递增链接程序还将有关对象文件、全局符号和重定位的信息以及填充部分保存在输出文件中，以便进行扩展。在后继的递增链接中，链接程序使用时间戳来确定哪些对象文件已更改，并更新这些模块。

在某些情况下，递增链接程序无法执行递增链接。如果出现这种情况，递增链接程序将自动执行初始递增链接来恢复该进程。在下列情况下，链接程序将自动执行输出文件的初始递增链接：

- 更改的链接程序命令行，其中链接程序命令行不匹配输出文件中存储的命令行（详细和跟踪选项例外）
- 填充空间全部用尽。
- 模块由 **ld -s** 或 **ld -x** 选项或工具（如 *strip*(1) ）进行了修改。
- 不兼容的递增链接程序版本，在早期版本创建的可执行文件上运行新版本递增链接程序时出现。

- 新工作目录，其中递增链接程序在当前目录更改时执行初始递增链接。
- 向（或从）链接程序命令行添加（或删除）归档或共享库。
- 对象文件已从链接程序命令行删除。

有关详细信息，请使用 **+help** 选项或参阅《Linker and Libraries User's Guide》。

Archive Library Processing

如果存在不满意的符号，递增链接程序将搜索归档库。它将提取所有使不满意符号满意的归档成员，并将其作为新对象文件进行处理。如果修改了归档库，链接程序将替换修改的归档库。

即使已删除对象文件中定义的符号的所有引用，从先前链接中的归档库中提取的对象文件仍保留在输出加载模块中。链接程序将在执行下一次初始递增链接时删除这些对象文件。

Shared Library Processing

在初始递增链接中，链接程序将扫描共享库符号表，并解析不满意的符号，其方式与在常规链接中相同。在递增链接中，链接程序根本不处理共享库及其符号表，并且不报告共享库不满意符号。不满意符号的检测由动态加载程序负责。如果修改了命令行上的任何共享库，链接程序将恢复到初始递增链接。

Performance

如果更改较大百分比的对象文件，则递增链接程序的性能会大大受到影响。

递增链接程序可能无法很快地链接小程序，而且可执行文件大小的相对增加大于较大程序大小的相对增加。

不要使用递增链接程序来创建最终的生产模块。由于它会保留附加的填充空间，由递增链接程序创建的模块要比在常规链接中创建的模块大得多。

外部语言环境影响

环境变量

LDOPTS

参数可以通过 **LDOPTS** 环境变量以及在命令行上传递给链接程序。链接程序获取 **LDOPTS** 的值，并将其内容放在命令行上任何参数之前。

LPATH

指定用于搜索库文件的缺省目录。请参阅 **-l** 选项。

LD_LIBRARY_PATH 和 **SHLIB_PATH**

在运行时指定用于搜索库文件的目录。有关详细信息，请参阅《Online HP-UX Linker and Libraries User's Guide》的 **-s** 选项和 **+help** 选项。

下列国际化变量会影响 **ld** 的执行：

LANG 当未提供 **LC_ALL** 和其他 **LC_*** 环境变量时，确定本国语言、本地惯例和编码字符集的语言环境类别。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值 **C** （请参阅 *lang*(5) ）而非 **LANG** 。

LC_ALL

确定所有语言环境类别的值，其优先级高于 **LANG** 和其他 **LC_*** 环境变量。

LC_MESSAGES

确定语言环境，用于影响写入标准错误中的诊断消息的格式与内容。

LC_NUMERIC

确定数字格式化时所用的语言环境类别。

LC_CTYPE

确定字符处理功能的语言环境类别。

NLSPATH

为处理 **LC_MESSAGES** 确定消息目录的位置。

如果任一国际化变量包含无效设置，则 **ld** 就会认为所有国际化变量均设置为 **C** 。请参阅 *environ*(5) 。

此外，下列环境变量也会影响 **ld** ：

TMPDIR

为临时文件指定一个目录（请参阅 *tmpnam*(3S) ）。

BROWSER

指定浏览器的路径名，以便在使用 **+help** 选项时显示《HP-UX Linker and Libraries Online User's Guide》。

诊断信息

ld 在链接成功时返回零。非零的返回代码指示已出错。

举例

链接 C 程序的一部分，以便稍后由 **ld** 进行处理（请注意输出对象文件的 **.o** 后缀；这是 HP-UX 指示可链接对象文件的惯例）：

```
ld -r file1.o file2.o -o prog.o
```

链接标准模式中的共享绑定程序。请注意，没有指定 **crt0.o** ，因为对于共享链接它不再是必需的。

```
ld himom.o -lc
```

链接简单的 Fortran 程序，以用于符号调试程序（请参阅 *wdb*(1) ）。由于未在命令行上指定 **-o** 选项，输出文件名是 **a.out** 。

```
ld ftn.o -lcl -lisamstub \
  -lc /opt/langtools/lib/pa20_64/end.o
```

创建共享库：

```
ld -b -o libfunc.so func1.o func2.o func3.o
```

使用内部名称创建共享库，该共享库允许动态库搜索：

```
ld -b -o libfoo1.so.1 foo1.o foo2.o +h libfoo1.so.1
ln -s libfoo1.so.1 libfoo1.so
cc -g mytest.c -L. -lfoo1
chatr a.out
...
 共享库列表:"
  libfoo1.so"
```

如果不使用 **+h** ，共享库将没有内部名称。链接程序不会检查 **.so** 是否是符号链接。如果共享库没有内部名称，则将记录所查看的库名。

```
chatr a.out
...
 共享库列表:
  libfoo1.sl
```

将 **$ORIGIN** 添加到共享库列表中的共享库名之前。

```
ld -dynamic main.o -L /usr/lib/hpux32/ +origin -lc
ld -dynamic main.o +origin /usr/lib/hpux32/libc.so
chatr a.out
...
 共享库列表:
  $ORIGIN/libc.so
```

警告

ld 认为某些名称具有特殊的含义。符号 **_end** 由链接程序保留，特指程序地址空间结尾之外的第一个地址。同样，符号 **_edata** 特指初始化数据之外的第一个地址，符号 **_etext** 特指程序文本之外的第一个地址。符号 **end** 、**edata** 和 **etext** 也由链接程序定义，但只有当程序包含对这些符号的引用并未定义这些符号时才如此定义（有关详细信息，请参阅 *end*(3C) ）。

链接程序将此处列出的任何符号的用户定义当作错误。

通过其选项，链接程序为用户提供了很大的灵活性。但是，直接调用链接程序的用户必须承担某些附加的责任。

无法保证链接程序将按照文件在库中的相同相对顺序从归档库中提取文件并将其包含在最终程序中。

一旦检测到任何兼容性问题，链接程序将发出警告。除此之外，其他问题包括体系结构问题以及可能在一段时间内更改的功能。其中一些问题包括：

- 链接程序对不满意符号的检查，它有时会跳过归档库中的某些对象文件。只有在同时提供 **-v** 选项时，才会发出该警告。

如 选项部分所述，本版本的链接程序不再支持某些选项。

链接程序接受以下选项，并发出警告消息。

- **-A** *name*
- **-C** *n*
- **-Fw**
- **-S**
- **+cg** *pathname*
- **+Oreusedir=***dir*
- **+OselectiveO3**
- **+Oselectivesize** *size*

以下选项支持兼容性。将接受并忽略它们:

- **-n**
- **-q**
- **-Fz**
- **-N**
- **-Q**
- **-T**
- **-V**
- **+cdp** *oldpath:newpath*
- **+gstbuckets** *size*
- **+nosmartbind**
- **+pdzero**
- **+plabel_cache** *flag*
- **+tools**
- **+vallcompatwarnings**
- **+v[no]compatwarnings**
- **+O[no]fastaccess**
- **+Ostaticprediction**

常规可执行文件的行为不匹配用 **+compat** 构建的可执行文件的行为。对于任何 **setuid** 或 **setgid** 程序， **dld** 将禁用通过环境变量 **SHLIB_PATH** 和 **LD_LIBRARY_PATH** 进行的任何动态库搜索。如果库仅存在于 **SHLIB_PATH** （或 **LD_LIBRARY_PATH** ）中指定的目录中，并且程序是 **setuid** 或 **setgid** 程序（即 uid 不等于 euid 或 gid 不等于 egid）且依赖于该库，则将收到“未找到库”运行时错误。只有当满足以下条件时， **dld** 才会使用动态路径查找（通过 **SHLIB_PATH** ）： **getuid () == geteuid () && getgid () == getegid ()** 。也就是说，如果 uid 或 gid 不匹配其有效对等项， **dld** 仅搜索记录的库路径。因此， dld 不会检查导致运行时错误“未找到库”的 **SHLIB_PATH** 。

作者

ld 由 AT&T、加州大学伯克利分校和 HP 联合开发。

文件

/usr/lib/hpux32/lib*	32 位系统归档和共享库
/usr/lib/hpux64/lib*	64 位系统归档和共享库
a.out	输出文件
/usr/lib/hpux32/dld.so	32 位动态加载程序
/usr/lib/hpux64/dld.so	64 位动态加载程序
/usr/ccs/lib/hpux32/crt0.o	32 位运行时启动文件
/usr/ccs/lib/hpux64/crt0.o	64 位运行时启动文件
/usr/lib/hpux32/milli.a	**ld** 自动搜索的 32 位 millicode 库
/usr/lib/hpux64/milli.a	**ld** 自动搜索的 64 位 millicode 库
/usr/lib/hpux64/millikern.a	ld 为嵌入系统自动搜索的 64 位 millicode 库
/usr/lib/nls/msg/$LANG/ld.cat	消息清单
/var/tmp/ld*	临时文件
flow.data	包含通过运行提供的可执行文件生成的配置文件数据的文件
/usr/ccs/bin/fdp	用于从用提供的可执行文件创建的配置文件数据库文件创建过程链接顺序的程序；由 **-P** 选项进行派生处理
/opt/langtools/bin/sin	用于提供可执行文件或共享库的静态提供程序。它在使用 **-I** 选项时调用。

另请参阅

配置处理和调试工具

adb(1)	绝对调试程序
gprof(1)	显示调用图形配置文件数据
prof(1)	显示配置文件数据
wdb(1)	C、C++、Fortran 符号调试程序

系统工具

aCC(1)	调用 HP-UX aC++ 编译程序
ar(1)	创建归档库
cc(1)	调用 HP-UX C 编译程序
chatr(1)	更改程序的内部属性
elfdump(1)	转储对象文件中包含的信息
exec(2)	执行文件
f90(1)	调用 HP-UX Fortran 90 编译程序

fastbind(1)	调用 fastbind 工具
lorder(1)	查找对象库的排序关系
nm(1)	输出对象文件的名称列表
size(1)	输出对象文件部分的大小
strip(1)	从对象文件删除符号和行编号信息

其他信息

a.out(4)	汇编程序、编译程序和链接程序输出
ar(4)	归档文件格式
crt0(3)	执行启动例程
dld.so(5)	动态加载程序
end(3C)	程序中最后位置的符号
uld.so(5)	微加载程序

文本和教程

《HP-UX Linker and Libraries Online User Guide》
（使用 **+help** 选项）

《HP-UX Linker and Libraries User's Guide》
（有关订购信息，请参阅 *manuals*(5)）

符合的标准

ld：SVID2、SVID3、XPG2、XPG4

名称

ld_pa： ld - 链接编辑器

概要

通用选项

ld [**-bdmnqrstvxzEGIOPQVZ**] [**-a** *search*] [**-c** *filename*] [**-dynamic**]
[**-e** *epsym*] [**-h** *symbol*]... [**-l***x* | *file*]... [**-l:** *library*]
[**-m**] [**-noshared**] [**-o** *outfile*] [**-u** *symbol*]... [**-y** *symbol*]...
[**-B** *bind*] [**-D** *offset*] [**-Fl** *lib*] [**-L** *dir*]... [**-R** *offset*] [**-Pd**]
[**-PD** *file*] [**-PF** *file*] [**+afs** *func_sym_x=func_sym_y*]...
[**+b** *path_list*] [**+compat**] [**+copyobjdebug**] [**+df** *file*] [**+e** *symbol*]
[**+ee** *symbol*] [**+fb**] [**+fbu**] [**+filter** *shared_library_path*]
[**+fini** *function*] [**+gst**] [**+gstsize** *size*] [**+h** *internal_name*] [**+help**] [**+init** *function*] [**+interp** *file*] [**+k**] [**+n**] [**+[no]mergeseg**] [**+[no]mmap**] [**+nocopyobjdebug**] [**+noobjdebug**] [**+objdebugonly**]
[**+origin** *shared_library_path*] [**+pd** *size*] [**+pgm** *name*] [**+pi** *size*] [**+s**] [**+std**] [**+tools**] [**+v[no]shlibunsats**] [**+vallcompatwarnings**] [**+v[no]compatwarnings**] [**+FP** *flag*] [**+I** *symbol*] [**+O[no]fastaccess**] [**+O[no]procelim**] [**+Oreusedir=**dir] [**+Oselectivepercent** *n*] [**+Oselectivesize** *size*] [**+OselectiveO3**] [**+Ostaticprediction**] [**+profilebucketsize** *size*] [**+allowdups**]

PA-RISC 32 位 (SOM) 选项

ld [**-NST**] [**-A** *name*] [**-C** *n*] [**-Fw**] [**-Fz**] [**+cdp** *oldpath:newpath*]
[**+cg** *pathname*] [**+dpv**] [**+ea** *filename*] [**+gstbuckets** *size*] [**+nosmartbind**] [**+plabel_cache** *flag*]

PA-RISC 64 位 (ELF) 选项

ld [**+allowrorelocs**] [**+[no]allowunsats**] [**+[no]forceload**] [**+hideallsymbols**] [**+ild**] [**+ildnowarn**] [**+ildpad** *percentage*] [**+ildrelink**] [**+interposer**] [**-k** *filename*] [**+[no]lazyload**] [**+linkersyms**] [**+nodefaultmap**] [**+noenvvar**] [**+nodynhash**]
[**+nosectionmerge**] [**+paddata** *pagesize*] [**+padtext** *pagesize*] [**+pdzero**]
[**+stripunwind**] [**+vtype** *type*]

说明

ld 将一个或多个对象文件或库作为输入，并将它们组合在一起生成单个文件（通常是可执行文件）。这样，它可以解析对外部符号的引用，将最终地址分配到过程和变量，修订代码和数据以反映新的地址（称作“重定位”的过程），并更新符号调试信息（当存在于文件中时）。缺省情况下， **ld** 将生成一个可执行文件，该文件可由 HP-UX 加载程序 **exec()** 运行（请参阅 *exec*(2) ）。或者，链接程序可以生成一个可重定位的文件，它适合由 **ld** 做进一步处理（请参阅下面的 **-r** ）。它也可生成一个共享库（请参阅下面的 **-b** ）。如果存在任何重复的符号或者仍然存在未解析的外部引用，链接程序会将输出文件标记为不可执行。如果在执行过程中出现其他任何错误， **ld** 可能会（也可能不会）生成输出文件（请参阅 **+k** 选项）。

ld 识别三种输入文件：由编译程序、汇编程序或链接程序生成的对象文件（也称作 **.o** 文件），由链接程序创建的共享库，以及对象文件的归档（称作归档库）。归档库包含其组件对象文件中所有外部可见符号的表。（归档程序命令 *ar*(1) 创建并维护该索引）。 **ld** 使用该表来解析对外部符号的引用。

ld 按照文件在命令行上出现的相同顺序对其进行处理。当且仅当该对象模块在用户程序或者共享库（或相关共享库）内提供当前未解析引用的定义时，它才会包括归档库元素中代码和数据。通常的做法是在命令行上所有简单对象文件名后列出库。

共享库中的代码和数据决不会复制到可执行程序中。如果程序使用共享库，32 位链接上的动态加载程序 **/usr/lib/dld.sl** 在启动时将由启动文件 **crt0.o** 调用。在 **/usr/ccs/lib/crt0.o** 或 **/opt/langtools/lib/crt0.o** 可找到 **crt0.o** 的完全相同副本。对于 64 位模式，动态加载程序是 **/usr/lib/pa20_64/dld.sl** ，由 exec 为使用共享库的程序调用。 **crt0.o** 在共享绑定链接中不是必需的。动态加载程序将每个必需的库附加到进程，并解析程序及其库之间的所有符号引用。共享库的文本段在使用该库的所有进程之间共享；使用该库的每个进程均会收到各自的数据段副本。如果 **pxdb -s on** 已经在加载库的可执行文件上运行，则将以私密的方式为运行该可执行文件的每个进程映射共享库的文本段。 **ld** 以递归方式检查由 **ld** 创建的程序所用的共享库的相关性。如果 **ld** 没有在共享库相关性列表中记录的路径处找到支持的共享库，并且相关性是创建共享库时使用的 **-l** 参数的结果， **ld** 会搜索将为使用 **-l** 指定的库而搜索的所有目录（请参阅 **-L** 和 **LPATH** ）。

环境变量

参数可以通过 **LDOPTS** 环境变量以及在命令行上传递给链接程序。链接程序获取 **LDOPTS** 的值，并将其内容放在命令行上任何参数之前。

LD_PXDB 环境变量定义调试预处理器 **pxdb** 的完整执行路径。缺省值是 **/opt/langtools/bin/pxdb** 。如果其输出文件是可执行文件并包含调试信息， **ld** 将对该文件调用 **pxdb** 。要将 **pxdb** 调用延迟到第一个调试会话，请将 **LD_PXDB** 设置为 **/bin/true** 。

LPATH 环境变量可用于指定用于搜索库文件的缺省目录。请参阅 **-l** 选项。

通用选项

首先列出通用 **ld** 选项，其后是仅在 32 位链接程序中支持的选项，然后列出仅在 64 位链接程序中支持的选项。

-a *search* 指定是否使用 **-l** 选项搜索共享库或归档库。 *search* 的值应该是 **archive** 、 **shared** 、 **archive_shared** 、 **shared_archive** 或 **default** 之一。该选项可以出现多次，在 **-l** 选项之间散布，以控制每个库的搜索。缺省设置是使用库的共享版本（如果可用）或者归档版本（如果不可用）。

如果 **archive** 或 **shared** 处于活动状态，则仅接受指定的库类型。

如果 **archive_shared** 处于活动状态，则首选归档形式，但允许使用共享形式。

如果 **shared_archive** 处于活动状态，则首选共享形式，但允许使用归档形式。

-b 创建一个共享库而不是常规的可执行文件。使用该选项处理的对象文件必须包含“位置无关代码” (PIC)。请参阅 *cc*(1) 、 *CC*(1) （可选 C++ 编译程序文档的一部分）、 *f77*(1) 、 *pc*(1) 、 *as*(1) 和《Linker and Libraries Online User Guide》中有关 PIC 的讨论。

-c *filename* 从文件中读取 ld 选项。每一行包含零个或更多个由空格分隔的参数。文件中的每一行（包括最后一行）必须以换行符结尾。 **#** 字符表示该行的其余部分是注释。要将 **#** 字符

转义，请使用序列 **##** 。

-d 强制“通用”存储的定义；即，为 **-r** 输出分配地址和大小。

-dynamic 它允许链接程序创建可使用共享库的程序。除非使用 **+compat** ，它是 64 位链接的缺省值。在 32 位模式下，如果链接行上没有共享库，链接程序将创建静态可执行文件。

-e *epsym* 将输出文件的缺省入口点地址设置为符号 *epsym* 的地址（该选项仅适用于可执行文件）。

-h *symbol* 在将符号表写入输出文件之前，将该名称标记为“本地”，使其不再在外部可见。这将确保将来由 **ld** 进行处理的过程中，该特定条目将不会与其他文件进行冲突。

可以指定多个 *symbol* ，但每个符号必须添加 **-h** 。如果在构建共享库或程序时使用，该选项将防止命名符号对于动态加载程序可见。

-l*x* 搜索库 **lib***x***.a** 或 **lib***x***.sl**, ，其中 *x* 是一个或多个字符。 **-a** 选项的当前状态确定是搜索库的归档 (**.a**) 版本还是共享 (**.sl**) 版本。由于在遇到库名时对库进行搜索，因此 **-l** 的放置位置非常重要。缺省情况下，32 位库位于 **/usr/lib** 和 **/usr/ccs/lib** 。64 位库位于 **/usr/lib/pa20_64** 。如果用户的环境中存在环境变量 **LPATH** ，它应该包含要搜索的目录的冒号分隔列表。将搜索这些目录（而不是缺省目录），但仍可以使用 **-L** 选项。如果程序使用共享库，动态加载程序 **/usr/lib/dld.sl** （32 位）或 **/usr/lib/pa20_64/dld.sl** （64 位）将尝试从链接时所在的相同目录中加载每个库（请参阅 **+s** 和 **+b** 选项）。

-l: *library* 搜索指定的库。与 **-l** 选项类似，但例外的是 **-a** 选项的当前状态不重要。库名可以是任意有效的文件名（请注意，先前版本要求库名包含前缀 **lib** 并以 **.a** 或 **.sl** 后缀结尾）。

-m 该选项在标准输出上生成加载映射。

-n 64 位 **ld** 会接受但将忽略该选项。生成文件类型为 **SHARE_MAGIC** 的可执行输出文件。这是缺省值。该选项与 **-N** 和 **-q** 不兼容。

-noshared 该选项强制链接程序创建完全归档绑定程序。

-o *outfile* 生成名为 *outfile* 的输出对象文件（如果未指定 **-o** *outfile* ，则为 a.out ）。

-q 64 位链接将忽略该选项。生成文件类型为 **DEMAND_MAGIC** 的可执行输出文件。该选项与 **-n** 、 **-N** 和 **-Q** 不兼容。

-r 为后继的重新链接将重定位信息保留在输出文件中。 **ld** 命令不报告未定义的符号。该选项不能在构建共享库 (**-b**) 时使用，也不能与 **-A** 或 **+ild** 递增链接选项一起使用。

-s 从输出文件中删除所有符号表、重定位和调试支持信息。这可能会妨碍或阻止对得到的程序使用符号调试程序或配置处理程序。该选项与 **-r** 不兼容（ *strip*(1) 命令也会删除这些信息）。该选项与 **+ild** 不兼容（递增链接需要用 **-s** 选项删除的输出加载模块部分）。

-t 在 **ld** 进行处理时输出每个输入文件的踪迹（到标准的输出）。

-u *symbol* 输入 *symbol* 作为符号表中的未定义符号。得到的未解析引用对于仅从库中的对象文件链接程序非常有用。可以指定多个 *symbol* ，但每个符号之前必须添加 **-u** 。

-v 在链接过程中显示详细消息。在 32 位系统上，对于加载的每个模块，链接程序会指示哪个符号导致该模块加载。对于 64 位系统，链接程序将仅为从归档库中加载的模块指示该信息。

-x 从输出文件中删除本地符号。这将减小输出文件的大小，而不会损害对象文件工具的有效性。该选项与 **-r** 选项不兼容。该选项与 **+ild** 选项不兼容。递增链接需要用 **-x** 选项删除的输出加载模块部分。

注意：使用 **-x** 可能影响调试程序或配置处理程序的使用。

-y *symbol* 指示出现 *symbol* 的每个文件。可以指定多个 *symbol* ，但每个符号之前必须添加 **-y** 。

-z 安排运行时取消引用的空指针来生成 **SIGSEGV** 信号（它是 **-Z** 选项的补充）。

-B *bind* 选择使用共享库的程序的运行时绑定行为或构建共享库时的绑定首选项。 *bind* 最常用的值包括：

direct 通过在符号解析过程中记录解析共享库的名称，创建符号引用和共享库之间的直接链接。该信息在运行时用于快速解析符号，而不必搜索当前加载的所有库。

-B direct 将隐式打开符号绑定（请参阅 **-B symbolic** ）并禁用相关的共享库处理。

通过设置 **LD_NODIRECTBIND** 环境变量，可以在运行时禁用直接绑定。

deferred
在第一个引用上（而不是在程序启动时）绑定地址。这是缺省值。

group 标记共享库，使其如同加载 **RTLD_GROUP** 标志以 **dlopen()** 一样进行操作。这不影响相关共享库。

注意：当前只有 64 位应用程序支持 **-B group** 绑定模式。

immediate
在加载库时立即绑定所有符号的地址。通常后接 **-B nonfatal** ，以便在第一个引用上解析在程序启动时无法解析的过程调用。

由于 **-B nonfatal** 禁止有关未解析符号的消息，也可以指定 **-B verbose** 显示这些消息。

请参阅下面的示例。

lazydirect

仅记录带有缓慢加载标记的共享库的直接绑定信息。请参阅 **+[no]lazyload** 。

nodelete

标记共享库，使得使用 **dlclose()** 或 **shl_load()** 的显式卸载静默地返回成功，而不将共享库与进程分离。因此，共享库句柄仅对于 **shl_findsym()** 有效。在使用 **shl_load()** 或 **dlopen()** 进行下一次显式加载之前，它对于 **dlsym()** 、 **dlclose()** 和 **shl_unload()** 一直无效。

nodirect

禁用直接绑定。仅记录“直接提示”，以引用带有缓慢加载标记的库。这是缺省行为。

nonfatal

如果还将 **-B immediate** 用于无法在程序启动时绑定的代码符号，请延迟绑定，直到引用它们。请参阅上面的 **-B immediate** 说明。

由于 **-B nonfatal** 禁止有关未解析符号的消息，也可以指定 **-B verbose** 显示这些消息。

restricted

使得符号定义搜索范围仅限于在库加载时可见的符号。

symbolic

该选项仅在构建共享库时使用（与 **-b** 选项一起），它使得库中所有未解析的符号在内部解析（如果可能）。缺省情况下，未解析的符号将解析为库内或库外最可见的定义。

verbose

在绑定符号时显示详细消息。这是缺省设置，但指定 **-B nonfatal** 时除外。这种情况下，必须显式指定 **-B verbose** 来获取详细消息。

有关如何使用绑定模式的详细信息，请参阅 **+help** 选项或《HP-UX Linker and Libraries User's Guide》手册。

-D *offset* 以十六进制形式设置数据段的原点。

与 **+ild** 选项一起使用时，如果在初始递增链接后更改偏移，链接程序将自动执行初始递增链接。

-E 标记程序定义要导出至共享库的所有符号。在 32 位链接或 **+compat** 模式 64 位链接中， **ld** 仅标记链接时所见的共享库实际引用的符号。在 64 位 **+std** 链接中，缺省情况下将导出所有符号，因此 **-E** 不一定使符号可见。但是，它有一种附加的副作用，即根据需要标识所有导出的符号，这样在使用无用代码删除 (**+Oprocelim**) 时将不会删除这些符号。

-Fl lib 强制加载归档库 *lib* 的所有成员对象。如果不使用 **-Fl** ，链接程序仅从 *lib* 中加载需要的归档对象。对于通过归档库创建共享库（如果成员对象是与位置无关的代码），或者当您需要加载定义共享库所需符号的归档成员时，该选项非常有用。

可以使用多个选项符号对在命令行上指定多个库，也就是说，您指定的每个库前必须添加 -Fl 选项。

-G 从输出文件中删除所有未加载的数据。该选项通常用于删除调试信息。

-I 提供用于在执行时收集配置文件信息的代码。在程序执行过程中收集的配置文件数据可以与 **-P** 选项一起使用。与该选项链接的 32 位程序应该使用启动文件 **/opt/langtools/lib/icrt0.o** 。该选项不应与 **-P** 、 **-A** 、 **-O** 、 **+ild** 或 **+O** 选项一起使用。

-L *dir* 在缺省位置中查找之前在 *dir* 中搜索 **lib***x***.a** 或 **lib***x***.sl** 。可以指定多个目录，但每个目录之前必须添加 **-L** 。 **-L** 选项只有在位于命令行上的 **-l** 选项之前时才有效。

-O 打开链接程序优化。优化目前包括从可执行文件的代码中删除不必要的 **ADDIL** 指令（仅限 32 位），以及删除无用的过程。

当选择 **+O4** 编译程序选项时， **-O** 将由编译程序传递给链接程序。

该选项与 **+ild** 选项不兼容。

有关链接程序优化的详细信息，请参阅 **+help** 选项或《HP-UX Linker and Libraries User's Guide》手册。

-P 检查提供的程序所生成的数据文件（请参阅 **-I** 选项），以执行基于配置文件的代码优化。该选项不应与 **-A** 或 **+ild** 选项一起使用。

-Q 对于 64 位链接忽略。生成文件类型为 **EXEC_MAGIC** 或 **SHARE_MAGIC** （取决于是指定 **-N** 还是 **-n** ）的可执行输出文件。这是缺省值。该选项与 **-q** 不兼容。

-R *offset* 以十六进制形式设置文本段（即代码段）的原点。

与 **+ild** 选项一起使用时，如果在初始递增链接后更改偏移，链接程序将自动执行初始递增链接。

-V 输出消息，提供有关所用 **ld** 版本的信息。

-Z 允许空指针的运行时取消引用。请参阅 *cc*(1) 中有关 **-Z** 和 *pointers* 的讨论（它是 **-z** 选项的补充）。

-Pd 将可调试函数重新排序。通常，由于重新排序会使包含调试信息的 **.o** 文件中的函数无法调试，因此 **-P** 不会将其重新排序。该选项将覆盖这一行为，将这些函数重新排序。

-PD *filename* 将 **fdp** 在链接过程中使用 **-P** 选项生成的链接顺序文件保存到用户指定的文件。该选项与 **+ild** 选项不兼容。

-PF *filename* 指示链接程序将指定文件用于链接程序文件，而不是使用 **/usr/ccs/bin/fdp** 生成该文件。该选项与 **+ild** 选项不兼容。

+afs "func_sym_x=func_sym_y ...

指示链接程序将函数符号替换为共享库和可执行文件链接中的备用函数符号。

+afs 选项支持函数符号别名。通常，用户程序包含与不同名的优化库函数的功能精确匹配的函数。通常会在程序中频繁调用这些用户定义的函数。使用 **+afs** 选项，可以通过在链接时将对用户定义函数的所有引用替换为对调整后的库函数的引用来大大提高性能，从而只需使用一个重新链接即可优化这些函数。

这两种函数都必须定义相同数量和类型的参数，并返回相同类型的值。如果它们不匹配，结果将不可预测，而链接程序不会生成警告消息。

例如：

$ **ld** ...**+afs func_sym1=func_sym2** ...

在该示例中。链接程序将对函数符号 *func_sym1* 的所有引用替换为对 *func_sym2* 的引用。 *func_sym2* 符号应该是常规的非别名符号。它不能出现在另一 **+afs** 选项上“=”的左侧。

可以使用多个选项符号对在命令行上指定多个函数符号别名，也就是说，您指定的每个符号对前面必须添加 **+afs** 选项。

+allowdups 允许多个符号定义。缺省情况下，在可重定位的对象之间出现的多个符号定义可能会导致致命的错误。该选项禁止该错误条件，并允许采用第一个符号定义。

+b *path_list* 指定要在程序运行时搜索的目录（嵌入路径）的冒号分隔列表，可定位使用 **-l** 或 **-l:** 选项指定的可执行输出文件所需的共享库。由单个冒号 **(:)** 组成的参数指示 **ld** 应该使用由 **-L** 选项和 **LPATH** 环境变量指定的所有目录来构建该列表（请参阅 **+s** 选项）。

+compat 32 位链接将忽略该选项。该选项打开链接程序中的兼容性模式 — 64 位链接模拟 32 位链接的行为。

+copyobjdebug 复制 objdebug 空间。

+df *file* 与 **-P** 选项一起使用时，该选项指定 *file* 应该用作配置文件数据库文件。缺省值为 **flow.data** 。有关详细信息，请参阅有关 **FLOW_DATA** 环境变量的讨论。该选项与 **+ild** 选项不兼容。

+e *symbol* 在构建共享库或程序时，标记符号以导出至动态加载程序。仅导出显式标记的符号。构建共享库时，将在内部解析未导出的符号调用。

+ee *symbol* 就导出符号而言，该选项与 **+e** 选项类似。但是，与 **+e** 选项不同， **+ee** 不更改文件中其他任何符号的可见性。在 32 位链接或 **+compat** 模式 64 位链接中，它具有导出指定符号而不隐藏在缺省情况下导出的任何符号的效果。在 64 位 **+std** 链接中，缺省情况下将

导出所有符号，因此 **+ee** 不一定使符号可见。但是，它有一种附加的副作用，即根据需要标识符号，这样在使用无用代码删除 (**+Oprocelim**) 时将不会删除该符号。当然，如果还给定了 **+hideallsymbols** 等选项， **+ee** 仍会保留其导出行为。

+fb 指示链接程序对它生成的可执行文件运行 fastbind 工具。该可执行文件应该与共享库链接。有关详细信息，请参阅 *fastbind*(1) 、 **+help** 选项或《HP-UX Linker and Libraries User's Guide》手册。该选项与 **+ild** 选项不兼容。

+fbu 将 **-u** 选项传递给 fastbind 工具。有关详细信息，请参阅 *fastbind*(1) 、 **+help** 选项或《HP-UX Linker and Libraries User's Guide》手册。该选项与 **+ild** 选项不兼容。

+filter *shared_library_path*

启用共享库过滤器机制，它可用于将较大的库拆分为一个“过滤器”和多个“实现”库，以便更有效地组织共享库。 *shared_library_path* 指定过滤器库的位置。有关详细信息，请参阅《HP-UX Linker and Libraries User's Guide》。

+fini *function_name*

指定将按照前进顺序（函数在命令行上从左到右的顺序）调用的终止程序函数。终止程序函数按照与初始设置的深度优先顺序相反的顺序调用。

在构建共享库、不完整的可执行文件或完全绑定的可执行文件时使用该选项。使用该选项指定的函数应该不采用任何参数，并且不返回任何内容（void 函数）。只有当 **main()** 调用 **exit()** 时，才调用终止程序。如果 **main()** 调用 **return()** ，则不调用终止程序。

不要将 **+fini** 与 **-r** 选项一起使用（链接程序会忽略 **+fini** 选项）。

可以使用多个选项符号对在命令行上指定多个终止程序函数，也就是说，您指定的每个函数前必须添加 **+fini** 选项。

有关终止程序函数的详细信息，请使用 **+help** 选项或参阅《HP-UX Linker and Libraries User's Guide》。

+gst 启用全局符号表散列机制，用于查找导入（或导出）条目的值。 **+gst** 和相关选项通过使用全局符号表（提高导出符号搜索的性能）提供了性能改进。有关详细信息，请参阅 *dld.sl*(5) 和《HP-UX Linker and Libraries Online User Guide》。

+gstsize *size* 使用全局符号表散列机制请求特定的散列阵列大小。缺省值为 1103 。通过将 **_HP_DLDOPTS** 环境变量设置为值 **-symtab_size** 质数，可以在运行时覆盖该值。可以使用 **chatr +gstsize** *size file* 设置该值。

+h *internal_name*

在构建共享库时，记录 *internal_name* 作为该库的名称。当库用于链接其他可执行文件（程序或共享库）时，该 *internal_name* 将记录在结果输出文件的库列表中，而不是输入共享库的文件系统路径名中。

也就是说，如果未使用 **+h** ，共享库将没有内部名称，并且当使用共享库构建可执行文件时，链接程序将记录它所查看的库名。

如果 *internal_name* 是完全限定名称，则将像是在它随后链接时所参照的任何可执行文件库列表中一样 (**as is**) 进行记录。如果 *internal_name* 是相对路径名，或者未指定任何目录组件，则 *internal_name* 将追加 (**appended**) 到它随后链接时所参照的任何可执行文件库列表中的输入共享库的文件系统目录组件。

如果在链接行上看到多个 +h 选项，将使用第一个并发出警告消息。

+help 启动帮助窗口工具《HP-UX Linker and Libraries Online User Guide》，它随某些 HP 编译程序附带（必须运行 X 窗口系统，并且 **DISPLAY** 环境变量必须设置为工作站或 X 终端的名称）。有关详细信息，请参阅《HP-UX Linker and Libraries User's Guide》手册。有关订购信息，请参阅 *manuals*(5) 。

+init *function_name*

指定将按照相反顺序（函数在命令行上从右到左的顺序）调用的初始设置函数。初始设置按照深度优先顺序调用。例如，当共享库加载时，将首先调用其所有相关库中的初始设置。

在构建共享库、不完整的可执行文件或完全绑定的可执行文件时使用该选项。使用该选项指定的函数应该不采用任何参数，并且不返回任何内容（void 函数）。只有当 **main()** 调用 **exit()** 时，才调用终止程序。如果 **main()** 调用 **return()** ，则不调用终止程序。

不要将 **+init** 与 **-r** 选项一起使用（链接程序会忽略 **+init** 选项）。

可以使用多个选项符号对在命令行上指定多个初始设置函数，也就是说，您指定的每个函数前必须添加 **+init** 选项。

有关初始设置函数的详细信息，请使用 **+help** 选项或参阅《HP-UX Linker and Libraries User's Guide》。

+interp *file* 更改 **dld** 路径，将该参数用作“解释程序”，而不是 **dld.sl** 。

+k 如果在链接中没有遇到错误，则指示链接程序仅创建可执行文件。如果发现错误（系统错误或未解析的引用），则将删除输出文件。

+mergeseg 启用共享库段合并功能。请参阅《HP-UX Linker and Libraries User's Guide》中有关共享库段合并的说明。通过它可以合并在程序启动时加载的共享库的所有数据段。这样，内核可以使用具有较大页大小的表条目，从而将提高运行时性能。

+mmap 启用该功能可使用 **mmap** 来写入输出数据。这是缺省行为。用 **mmap** 启用的链接程序要快得多。

+n 使得链接程序在搜索任何归档或共享库之前加载所有对象模块。然后，链接程序将按从左到右的顺序搜索在命令行上指定的归档和共享库。在命令行上从左到右重复搜索库，直到再也没有不满足的符号，或者最后的搜索没有添加新的定义。如果指定了两个具有

符号相关性的库，该选项将非常有用。

+nocopyobjdebug

不复制 objdebug 空间。使用该选项（与链接行上的 **-r** 对象文件一起）可禁止将 LINKMAP 空间复制到可执行文件的缺省行为。

+noobjdebug 覆盖 **+objdebug** 编译程序选项，并将所有调试信息复制到可执行文件。

与 **-g** 一起使用时， **+objdebug** 会将调试信息保留在对象文件中，而不是在链接时将其复制到可执行文件，这将缩短链接时间并减小可执行文件的大小。编译时缺省值 **+noobjdebug** 将调试信息复制到可执行文件。

如果在编译时指定 **-g** ，编译程序会将符号调试信息放入对象文件。缺省情况下，链接程序将调用 **pxdb** ，它将压缩该调试信息，并将其复制到可执行文件。如果在编译时使用 **+objdebug** ，链接程序会将调试信息保留在对象文件中。要调试可执行文件，HP WDB 调试程序必须具有对对象文件的访问权。如果移动对象文件，请使用 HP WDB 的 **objdir** 命令指示对象文件的位置。通过避免这样复制调试信息， **+objdebug** 选项可减少链接时间和可执行文件的大小。

编译时缺省值是 **+noobjdebug** 。如果链接程序检测到任何用 **+objdebug** 编译的对象文件，则会将调试信息保留在这些文件中。对于未用 **+objdebug** 编译的所有对象文件，均会将其调试信息复制到可执行文件。可以将调试信息保留在某些对象文件而不是其他对象文件中。

如果归档成员未使用 **+objdebug** 编译（并且它们包含调试信息，即它们是使用 **-g** 编译的），归档成员中的调试信息将在缺省情况下复制到 **a.out** 。链接程序不会将调试程序从共享库复制到 **a.out** 中，无论它们是使用 **+objdebug** 还是 **+noobjdebug** 编译的。

当进行链接以显式指示链接程序将所有调试信息复制到可执行文件（即使是从用 **+objdebug** 编译的文件中）时，请使用 **+noobjdebug** 选项。

+nomergeseg 禁用共享库段合并功能。这是缺省值。

+nommap 禁用 **mmap** 功能，并使用常规缓冲方法写入输出文件。 请注意：除非出现系统内存不足并且链接程序因此而失败的情况下，否则不要使用 **+nommap** 。

+objdebugonly 忽略非 objdebug 对象或归档中的调试信息，并继续在 **+objdebug** 模式下操作。该选项可以从 C 或 C++ 编译程序命令行中作为 **-Wl,+objdebugonly** 进行传递。如果仅调试使用 **+objdebug** 选项编译的文件， **+objdebugonly** 可以指示链接程序绕过处理用 **+noobjdebug** 编译的文件中的调试信息的步骤。通过 **+objdebugonly** 选项，链接程序将禁止对 **pxdb** 的调用。

+origin shared_library_name -lx

（仅在 **-l** 选项或共享库名之前使用）。使链接程序将 **$ORIGIN** 添加到共享库列表中的共享库名之前。在运行时，动态加载程序将确定父模块（共享库或可执行文件）的当前目录，然后将 **$ORIGIN** 替换为该目录的名称。

例如:

$ **ld -dynamic main.o +origin libx.sl -L /usr/lib/ -lc**

当 **+origin** 选项可用时，建议使用 **+b** 选项在嵌入路径中指定 **$ORIGIN** ，例如:

$ **ld -dynamic main.o -L /usr/lib/ -lc +B $ORIGIN**

有关 $ORIGIN 的详细信息，请使用 +help 选项或参阅《HP-UX Linker and Libraries User's Guide》。

+pd *size* 请求应该用于数据的特定虚拟内存页大小。支持的大小包括 **4K** 、 **16K** 、 **64K** 、 **256K** 、 **1M** 、 **4M** 、 **16M** 、 **64M** 、 **256M** 、 **D** 和 **L** 。大小 **D** 允许内核选择应该使用的页大小。大小 **L** 会使用可用的最大页大小。如果无法满足请求的大小，则实际的页大小可能有所不同。

+pgm *name* 与 **-P** 选项一起使用时，该选项指定 *name* 应该用作配置文件数据库文件中的查找名称。缺省值是输出文件的基名（由 **-o** 选项指定）。该选项与 **+ild** 选项不兼容。

+pi *size* 请求应该用于指令的特定虚拟内存页大小。有关其他信息，请参阅 **+pd** 选项。

+s 指示在运行时，共享库加载程序可以使用环境变量 **SHLIB_PATH** 和 **LD_LIBRARY_PATH** （仅限 64 位）定位用 **-l** 或 **-l:** 选项指定的可执行输出文件所需的共享库。环境变量应该设置为目录的冒号分隔列表。如果同时使用了 **+s** 和 **+b** ，它们在命令行上的相对顺序将指示首先搜索哪个路径列表（请参阅 **+b** 选项）。

+std 32 位链接将忽略该选项。打开标准模式，这是 64 位模式下缺省值。使用该选项打开的选项包括: **-dynamic** 。指定该选项时关闭或忽略的选项包括: **+compat**、**+noenvvar**、**-noshared**。

+vallcompatwarnings

64 位 **ld** 会接受但将忽略该选项。显示有关任何兼容性问题相关警告的详细信息。缺省情况下，仅输出简洁消息。有关其他信息，请参阅下面的 “警告”部分。

+v[no]compatwarnings

64 位 **ld** 会接受但将忽略该选项。启用 [禁用] 输出有关系统间兼容性问题的警告。其中包括可能在将来发布版中更改的任何功能。缺省值是 **+vcompatwarnings** 。有关其他信息，请参阅下面的 “警告”部分。

+v[no]shlibunsats

启用 [禁用] 打印共享库所用的不满足的符号的列表。缺省值是 **+vnoshlibunsats** 。由于不使用引用符号的模块，因此链接程序所报告的某些不满足的符号在运行时是不必要的。

+FP *flag* 指定在程序启动时应该如何初始化浮点运算的环境。缺省情况下将禁用所有行为。支持以下标志（启用大写标志；禁用小写标志）:

V (**v**) 限制无效的浮点运算

Z (**z**) 限制除以零

O (**o**) 限制浮点上溢

U (**u**) 限制浮点下溢

I (**i**) 限制产生不精确结果的浮点运算。

D (**d**) 启用去标准化值的突然下溢（下溢到零）。

注意：启用突然下溢是基于 PA-RISC 1.0 的系统上的未定义操作，但是它在 PA-RISC 的所有后继版本上定义。只有当该标志在运行时所用的处理器上可用时，选择该标志才会启用突然下溢。

要在运行时动态更改这些设置，请参阅 *fesettrapenable(3M)* 。

+I *symbol* 指定构建共享库时初始设置函数的名称。一个共享库可以指定多个初始设置。初始设置按照其指定顺序执行。可以指定多个初始设置，但是每个初始设置前必须添加 **+I** 选项。有关初始设置函数的详细信息，请参阅 **+help** 选项或《HP-UX Linker and Libraries User’s Guide》手册。

+O[no]fastaccess

启用 [禁用] 对全局数据的快速访问。链接程序会重新排列数据，以进一步减少可执行文件中 **ADDIL** 指令的数量。

这一优化可能会指示与数据布局假定相关的一些细微编程错误。该优化可以在优化级别 2、3 和 4 上发生。缺省设置是优化级别 2 和 3 上的 **+Onofastaccess** 和优化级别 4 上的 **+Ofastaccess** 。

如果在存在调试信息的情况下使用 **+Ofastaccess** 选项，则将生成错误消息，并可能导致调试信息损坏。避免将 **+Ofastaccess** 与 **-g** 选项一起使用。

64 位 **ld** 会接受但将忽略该选项。

+O[no]procelim 启用 [禁用] 删除应用程序不引用的过程。缺省值是 **+Onoprocelim** 。过程删除可能会在任意优化级别（包括级别 0）上发生。有关详细信息，请参考 **+help** 选项或《HP-UX Linker and Libraries User’s Guide》手册。该选项与 **+ild** 选项不兼容。

+Oreusedir=*dir* 指定目录的名称，以用作对象代码复用功能的复用数据库。该目录包括要复用的已编译对象文件。 *dir* 可以是在其中调用链接程序的目录的绝对或相对路径。 **+Oreusedir** 选项在使用 **+O4** 或基于配置文件的优化时不将中间对象代码重新编译成对象代码，从而将缩短链接时间。

+Oselectivepercent *n*

指示过程内优化程序驱动程序将对象文件的前百分之 *n* 传递到高级别的优化程序进行过程内优化（如内联）。

+Oselectivesize *size*

指示过程内优化程序驱动程序将前 k 个例程传递到高级别的优化程序进行过程内优化，其中 k 个例程的大小接近但小于 *size* 。

+OselectiveO3 指示过程内优化程序驱动程序编译在 +O3 编译的 +O4 列表中不包括的例程。

+Ostaticprediction

该选项仅在 PA-RISC 2.0 体系结构上才有意义，它设置输出可执行文件的辅助标头中的分支预测位。

+profilebucketsize *size*

控制配置处理计数器的大小。该变量的可接受值为 16 或 32。如果指定的值不是 16 或 32，则将发出警告。该计数器的缺省值是 16，该值在未指定有效值时使用。链接时的值可以使用 **PROFILE_BUCKET_SIZE** 环境变量覆盖。有关详细信息，请参阅 *gprof*(1) 中的说明。

PA-RISC 32 位选项

-A *name*

该选项指定递增加载。链接的排列方式使得可以将得到的对象读入已经执行的程序。参数 *name* 指定一个文件，该文件的符号表提供了定义其他符号的基础。只有新链接的资料才会输入 **a.out** 的文本和数据部分，但新的符号表将反映在递增加载之前和之后定义的所有符号。此外， **-R** 选项可以与 **-A** 一起使用，允许新链接的段在相应的地址开始。缺省的起始地址是旧的 **_end** 值。**-A** 选项与 **-r** 和 **-b** 不兼容。另请注意，用 **ld -A** 动态加载代码的程序不能使用共享库。有关该选项的说明，请参阅 **+help** 选项或《HP-UX Linker and Libraries User's Guide》手册。

-C*n* 将最大参数检查级别设置为 *n* 。缺省最大值为 3。有关参数检查级别的含义，请参阅语言手册。

-Fw 不发出 unwind 表。如果编译程序或工具需要 unwind 表，请不要使用该选项。

-Fz 禁用将对 **$$dyncall_external** 某些调用转换成对 **$$dyncall** 的调用的链接程序功能。

-N 生成文件类型为 **EXEC_MAGIC** 的可执行输出文件。该选项与 **-n** 和 **-q** 不兼容。该选项使得数据直接放置在文本之后，并使文本可写。这种文件不能共享。

-S 为输出文件生成初始程序加载程序 (IPL) 辅助标头，而不是缺省的 HP-UX 辅助标头。

-T 在链接时将加载数据和重定位信息保存在临时文件而不是内存中。该选项将减少链接程序的虚拟内存要求。如果已设置 **TMPDIR** 环境变量，则在指定的目录（而不是在 **/var/tmp** ）中创建临时文件。

+cdp *oldpath:newpath*

替换 **a.out** 中共享库的记录路径。在 32 位模式下， **ld** 在链接时搜索的所有共享库的绝对路径名记录在 **a.out** 文件中。当程序开始执行时，动态加载程序将附加在链接时搜索的所有共享库。虽然可以使用 **+b** 和（或） **+s** 链接程序选项将动态加载程序定向到搜索共享库的目录，作为最后手段，动态加载程序将在 **a.out** 中的绝对记录路径中搜索共享库。可以指定多个共享库 *oldpath:newpath* ，但每个路径前面必须添加 **+cdp** 选项。

+cg *pathname*

指定使用 *pathname* 作为将 ISOM 编译成 SOM 的代码生成器。有关详细信息，请参阅《HP-UX Linker and Libraries Online User's Guide》中有关基于配置文件的优化的讨论。

+dpv 显示有关因无用过程删除而删除的过程的详细信息。将显示已删除过程的符号名、输入对象文件和字节大小。还将显示已删除过程的总字节大小。

+ea *filename*

导出文件 *filename* 中的所有符号。

+gstbuckets *size*

使用全局符号表散列机制请求每个条目的特定容器数。缺省值为 3。通过将 **_HP_DLDOPTS** 环境变量设置为值 **-symtab_buckets** *number* ，可以在运行时覆盖该值。可以使用 **chatr +gstbuckets** *size file* 设置该值。

+nosmartbind

在绑定共享库时禁用 SmartBind。启用该选项后，链接程序会将链接中的所有符号放入单个 SmartBind 模块，而不是将每个 **.o** 文件放入其自己的模块。

+plabel_cache

启用 PLABEL 缓存机制。将该选项与 **+gst** 选项一起使用。

该选项仅用于 C++。在 C++ 应用程序中，动态加载程序需要重复访问 PLABEL 信息（导入 Stub）。为了加快该访问的速度，动态加载程序使用全局符号表结构同时包含 PLABEL 条目。在 dl_header 结构中设置 PLABEL_CACHE 标志时，将启用该行为（已启用 **ld +plabel_cache enable a.out** 或 **chatr +plabel_cache enable a.out**））。

PA-RISC 64 位选项

+allowrorelocs （仅限内核虚拟环境支持）。允许在只读部分（如 **.text** 和 **.rodata** ）中生成动态重定位。通常，只读部分中需要动态重定位时，链接程序将发出错误消息。有关详细信息，请参阅内核虚拟环境支持。

+[no]allowunsats

+allowunsats 在得到的输出文件包含不满足的符号时不标志错误。这是可重定位链接和共享库版本的缺省值。 **+noallowunsats** 在得到的输出文件包含不满足的符号时标志错误。这是程序文件的缺省值。

+[no]forceload **+forceload** 加载归档库中的所有对象文件。 **+noforceload** 是缺省值 — 仅从归档库中加载所需的对象文件。在更改之前，显式或缺省选择的模式将一直有效。

+hideallsymbols 该选项用于防止所有符号的导出，除非使用 **+e** 显式导出。

+ild 指定递增链接。

如果不存在输出文件，或者输出文件不是使用 **+ild** 选项创建的，链接程序将执行初始递增链接。所生成的输出文件适用于后继的递增链接。递增链接选项对于可执行文件和共

享库链接均有效。

请注意： HP WDB 符号调试程序仅支持调试在启用 **+objdebug** 编译程序选项时创建的递增链接加载模块。（HP DDE 调试程序不支持 **+ild** 选项）。

要调试递增链接库，应使用 **+ild** 和 **+objdbg** 将其编译并链接。

如果指定以下与 **+ild**
不兼容的 ld 选项之一，链接程序将发出警告消息并忽略 **+ild** 选项。

-r 创建可重定位的对象文件。

Strip options:

-s 和 **-x** 删除输出文件的内容（递增链接需要用这些选项删除的输出加载模块部分）。

Optimization options:

-I 、 **-O** 、 **-P** 、 **-PD** 、 **-PF** 、 **+df** *file* 、 **+fb** 、 **+fbu** 、 **+fbs** 、 **+pgm** *name* 、 **+Oprocelim**

以下选项在一定限制下与 **+ild** 选项兼容:

-D offset , -R offset

设置数据段和文本段的原点。如果在初始递增链接后更改偏移，链接程序将自动执行初始递增链接。

-k *mapfile*

提供非缺省的映射文件。用户指定的映射文件规范可以使用 **+ild** 选项来启用。但不应该在初始递增链接后修改映射文件。如果在初始链接后修改映射文件，则将自动修改初始递增链接。

+ildnowarn 禁止递增链接相关警告。缺省情况下，链接程序将发出所有递增链接相关警告。如果在没有 **+ild** 或 **+ildrelink** 的情况下使用该选项，则会将其忽略。

+ildpad *percentage*

控制递增链接程序相对于填充的对象文件结构的大小所分配的填充 *percentage* 。缺省情况下，链接程序将分配填充空间的 25%。如果在没有 **+ild** 或 **+ildrelink** 的情况下使用该选项，则会将其忽略。

+ildrelink 执行初始递增链接，而不管输出加载模块。

在某些情况（如内部填充空间已用尽）下，将强制递增链接程序执行初始递增链接。**+ildrelink** 选项可用于通过定期重建输出文件来避免这些意外的初始递增链接。

+interposer 仅在构建共享库时使用。它将创建一个可用于内插的共享库。当使用直接绑定信息解析应用程序引用（请参阅 **-B direct** ）时，动态加载程序将首先搜索内插库。如果符号无法解析为任何内插库，则将使用直接绑定信息。

-k *filename* 指定对输出文件内存映射进行描述的映射文件。

用户指定的映射文件规范可以使用 **+ild** 选项来启用。但不应该在初始递增链接后修改映射文件。如果在初始链接后修改映射文件，则将自动修改初始递增链接。

有关详细信息，请参阅《HP-UX Linker and Libraries User's Guide》指南。另请参阅 **+nodefaultmap** 。

+[no]lazyload 启用 [禁用] 共享库的慢速加载。指定为 +lazyload 的共享库加载将延迟到在执行过程中对其进行引用时。

将不会在链接时处理 +lazyload 共享库的相关库，除非在链接行上显式指定这些库。

通过设置 **LD_NOLAZYLOAD** 环境变量，可以在运行时禁用慢速加载。

+linkersyms （仅限内核虚拟环境支持）。即使构建共享库，仍将创建链接程序生成的符号。支持虚拟环境的内核使用 **-b** 选项构建，该选项在缺省情况下不在共享库中创建链接程序生成的符号。有关详细信息，请参阅内核虚拟环境支持。

+nodefaultmap 该选项指示链接程序不使用缺省的内存映射。用户需要通过 **-k** 链接程序选项提供一个映射文件。

+nodynhash 禁用 **+gst** 选项的缺省链接程序行为，以创建可执行文件或共享库的 .dynhash 部分。使用该选项可禁止为以下库或可执行文件生成预计算的散列表信息：很少通过全局符号表查找方案使用，或者存储预计算散列值的系统开销太大的库或可执行文件。与 **-r** 选项一起使用时，该选项不起作用。

+noenvvar 指示动态加载程序不要在运行时查看 **LD_LIBRARY_PATH** 、 **SHLIB_PATH** 和 **$ORIGIN** 环境变量。如果指定了 **ld +compat** ，将启用该选项。（但是，动态加载程序会将 **LD_PRELOAD** 环境变量与 **+noenvvar** 选项一起使用）。在缺省情况下或者当指定 **ld +std** 时，将禁用该选项。请参阅 **+compat** 或 **+std** 。该选项通常用于安全程序（如 setuid）。

+nosectionmerge
与 **-r** 选项一起使用时可允许过程独立定位。缺省设置是将所有过程合并到单个位置。

+paddata *pagesize*
用零将数据段填充到 *pagesize* 的倍数。这可以改进页分配，从而提高 TLB 命中率。

+padtext *pagesize*
用零将文本段填充到 *pagesize* 的倍数。这可以改进页分配，从而提高 TLB 命中率。

+pdzero 在数据段的开头生成 4K 字节的零页。它还会将数据段的缺省起始地址设置为 0x8000000000000000。该选项允许内核分配较大的数据页，从而可提高 TLB 命中率和性能。

+stripunwind 不要输出 unwind 表。这将创建较小的可执行文件大小。如果不需要使用 unwind 表进行调试或 C++ 异常处理，请使用该选项。

+tools 请求链接应用程序，以使用 CXperf 进行配置处理。

+vtype *type* 生成有关链接操作的详细输出。 *type* 可以具有以下值：

files 转储有关所加载的每个对象文件的信息。

heap 转储有关链接所用堆大小的信息。

libraries
转储有关所搜索的库的信息。

procelim
转储有关已经由 **+Oprocelim** 选项删除的部分的信息

sections
转储有关添加到输出文件的每个输入部分的信息。

symbols
转储有关在输入文件中引用（或定义）的全局符号的信息。

all 转储上述所有信息。与 **-v** 相同。

缺省值

除非另行指定， **ld** 会将其输出文件命名为 **a.out** 。 **-o** 选项可覆盖该缺省值。可执行输出文件的类型为 **SHARE_MAGIC** 。 **-a** 的缺省状态是搜索共享库（如果可用）或归档库（如果共享库不可用）。缺省的绑定行为是 **deferred** 。

-Z/-z 选项的缺省值是 **-Z** 。

对于 64 位模式， **+std** 在缺省情况下启用。

使用 **ld** 进行递增链接（仅限 **64** 位模式）

在编辑-编译-链接-调试开发周期中，链接时间是一种重要的组成部分。递增链接程序（可通过 **+ild** 和 **+ildrelink** 选项使用）利用了可重复使用程序先前版本的大部分内容并且不需要处理未更改的对象文件等事实，可以减少链接时间。通过递增链接程序，可以将对象代码插入先前创建的输出文件（可执行文件或共享库）中，而不必重新链接未修改的对象文件。在初始递增链接后进行重新链接所需的时间取决于所修改的模块的数量。

链接程序可以执行以下不同的链接模式：

- 常规链接：链接程序链接所有模块的缺省操作模式。
- 初始递增链接：如果请求递增链接时，不存在递增链接程序创建的输出模块，或者该模块存在，但递增链接程序无法执行递增更新，则进入此模式。
- 递增链接：如果请求递增链接时，存在递增链接程序创建的输出模块，并且递增链接程序不需要初始递增链接，则进入该模式。

递增链接通常比常规链接快得多。在初始链接上，递增链接程序所需的时间与常规链接进程所需的时间大致相同，但是后继递增链接可以比常规链接快得多。在大小中等的链接（数十个文件，总大小为几个 MB）中，一个对象文件中的更改通常比常规 **ld** 链接快 10 倍。递增链接程序可以执行与分配填充空间及其他约束条件所许可的一样多的递增链接。链接时间降低的代价是可执行文件或共享库大小的增加。

递增链接程序为输出文件的所有组件分配填充空间。填充可使模块变得比那些通过 **ld** 链接的模块更大。随着对象文件的大小在连续递增链接过程中不断增加，递增链接程序可能会耗尽可用的填充空间。如果出现这种情况，将显示警告消息，并执行该模块的完整初始递增链接。当对象文件发生更改时，递增链接程序不但在所链接的可执行文件或共享库中替换该文件的内容，而且调整对该对象文件中定义的且由其他对象引用的所有符号的引用。这通过查看在递增链接的可执行文件或共享库中保存的重定位记录来实现。

在初始递增链接上，链接程序按照常规链接的相同方式处理输入对象文件和库。除了常规链接进程之外，递增链接程序还将有关对象文件、全局符号和重定位的信息以及填充部分保存在输出文件中，以便进行扩展。在后继的递增链接中，链接程序使用时间戳和文件大小来确定哪些对象文件已更改，并更新这些模块。

在某些情况下，递增链接程序无法执行递增链接。如果出现这种情况，递增链接程序将自动执行初始递增链接来恢复该进程。在下列情况下，链接程序将自动执行输出文件的初始递增链接：

- 更改的链接程序命令行，其中链接程序命令行与输出文件中存储的命令行不匹配（详细和跟踪选项例外）
- 填充空间全部用尽。
- 模块由 **ld -s** 或 **ld -x** 选项或工具（如 **strip**(1)）进行了修改。递增链接需要用这些选项删除的输出加载模块部分。
- 不兼容的递增链接程序版本，在早期版本创建的可执行文件上运行新版本递增链接程序时出现。
- 新工作目录，其中递增链接程序在当前目录更改时执行初始递增链接。
- 向（或从）链接程序命令行添加（或删除）归档或共享库。
- 对象已向（或从）链接程序命令行添加（或删除）。

有关详细信息，请参阅 《*Online Linker and Libraries User’s Guide*》 (**ld +help**)。

Archive Library Processing

如果存在不满足的符号，递增链接程序将搜索归档库。它将提取所有使不满足的符号满足的归档成员，并将其作为新对象文件进行处理。如果修改了归档库，链接程序将替换修改的归档库。

即使已删除对象文件中定义的符号的所有引用，从先前链接中的归档库中提取的对象文件仍保留在输出加载模块中。链接程序将在执行下一次初始递增链接时删除这些对象文件。

Shared Library Processing

在初始递增链接中，链接程序将扫描共享库符号表，并解析不满足的符号，其方式与在常规链接中相同。在递增链接中，链接程序根本不处理共享库及其符号表，并且不报告共享库不满足符号。动态加载程序将在运行时检测

它们。如果修改了命令行上的任何共享库，链接程序将恢复到初始递增链接。

Performance

如果更改了大多数对象文件，则递增链接程序的性能会大大受到影响。

递增链接程序可能无法很快地链接小程序，而且可执行文件大小的相对增长要大于较大程序的可执行文件大小的相对增长。

通常，链接程序需要扫描链接行上的所有共享库，以确定所有不满足的符号（即使是在递增链接中）。该进程可能会减慢递增链接的速度。递增链接程序不扫描共享库，而将共享库不满足符号的检测留给动态加载程序处理。

不要使用递增链接程序来创建最终的生产模块。由于它会保留附加的填充空间，由递增链接程序创建的模块要比在常规链接中创建的模块大得多。

请注意：

HP WDB 符号调试程序仅支持调试在启用 **+objdebug** 编译程序选项时创建的递增链接加载模块。（HP DDE 调试程序不支持 **+ild** 选项）。

修改可执行文件的任何程序（如 **strip**(1)）均可能会影响 **ld** 执行递增链接的能力。如果出现这种情况，递增链接程序将发出消息，并执行初始递增链接。

处理对象文件的第三方工具可能会在递增链接程序生成的模块上产生意外的结果。

内核虚拟环境支持

要为内核提供虚拟环境支持，请使用以下链接程序选项（而不是 **-noshared** 选项）构建内核：

$ **ld** ...**-Bsymbolic -b +hideallsymbols +allowrorelocs +linkersyms** ...

使用这些选项，可以将内核创建为不包含导出符号的独立共享库（使用 **-Bsymbolic** 、 **-b** 和 **+hideallsymbols** 选项）。由于内核包括对链接程序生成符号（如 **_etext** 、 **_end** 和 **_edata** ）的引用， **+linkersyms** 选项将指示链接程序创建这些符号（在缺省情况下不为共享库创建这些符号）。内核包含直接调用和只读数据空间中的重定位。它们通常在链接时应用，并且修正到链接时虚拟地址。对于可重定位的内核，链接程序需要创建动态重定位，在运行时修正这些引用。 **+allowrorelocs** 选项指示链接程序允许在只读部分进行动态重定位（链接程序通常会发出错误消息）。

外部语言环境影响

环境变量

下列环境变量会影响 **ld** 的执行：

BROWSER

指定 HTML 浏览器的路径名，以便在使用 **+help** 选项时显示《HP-UX Linker and Libraries Online User's Guide》。

ENABLE_PBO_FORK

缺省情况下，当提供的可执行文件（请参阅 **-I** 选项）写入配置文件数据时，它将创建一个在后台运行的单独进程；主进程将退出。如果该环境变量设置为“OFF”，则不创建单独的进程，但相同的进程会写

出配置文件数据。执行可执行文件后，如果在将其删除或修改时收到 ETXTBSY 错误，该选项将非常有用。该环境变量仅对于 32 位可执行文件有效。对于提供的共享库，配置文件数据始终由同一个进程来写入。

FDP_FORK
其功能与 **ENABLE_PBO_FORK** 相同，但用于 64 位可执行文件。

FLOW_DATA
提供的可执行文件（请参阅 **-I** 选项）将配置文件数据写出到当前目录中名为 flow.data 的数据库文件中。该文件的名称和位置可以通过将 FLOW_DATA 设置为所需的路径名来指定。配置文件数据以一个查找名称存储在数据库文件中，该名称与在运行时指定的可执行文件的基名相同。单个 flow.data 文件可保存配置文件数据

LDOPTS
参数可以通过 **LDOPTS** 环境变量以及在命令行上传递给链接程序。链接程序获取 **LDOPTS** 的值，并将其内容放在命令行上任何参数之前。

LPATH
指定用于搜索库文件的缺省目录。请参阅 **-l** 选项。

LD_LIBRARY_PATH、 **SHLIB_PATH** 和 **LD_PRELOAD**
在运行时指定用于搜索库文件的目录。有关详细信息，请参阅《Online HP-UX Linker and Libraries User's Guide》的 **+s** 选项、 *dld.sl*(5) 和 **+help** 选项。

LD_PROFILE
在运行时指定将进行配置处理的共享库的路径（有关详细信息，请参阅“HP-UX Linker and Libraries Online User's Guide”）。

LD_GPROF_LIB_NAME
在运行时指定用于对共享库进行配置处理的配置处理程序的路径。缺省值是 /usr/lib/libgprof32.sl （32 位应用程序）或 /usr/lib/pa20_64/libgprof.sl （64 位应用程序）。

TMPDIR
指定临时文件的目录。

下列国际化变量会影响 **ld** 的执行:

LANG 当未提供 **LC_ALL** 和其他 **LC_*** 环境变量时，确定本国语言、本地惯例和编码字符集的语言环境类别。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值 **C** （请参阅 *lang*(5) ）而非 **LANG** 。

LC_ALL
确定所有语言环境类别的值，其优先级高于 **LANG** 和其他 **LC_*** 环境变量。

LC_MESSAGES
确定语言环境，用于影响写入标准错误中的诊断消息的格式与内容。

LC_NUMERIC
确定数字格式化时所用的语言环境类别。

LC_CTYPE
确定字符处理功能的语言环境类别。

NLSPATH
为处理 **LC_MESSAGES** 确定消息目录的位置。

如果任一国际化变量包含无效设置，则 **ld** 就会认为所有国际化变量均设置为 **C** 。请参阅 *environ*(5) 。

诊断信息

ld 在链接成功时返回零。非零的返回代码指示已出错。

举例

链接 C 程序的一部分，以便稍后由 **ld** 进行处理（请注意输出对象文件的 **.o** 后缀；这是 HP-UX 指示可链接对象文件的惯例）：

```
ld -r file1.o file2.o -o prog.o
```

在 32 位模式下，链接简单的 FORTRAN 程序，以用于 **dde** 符号调试程序（请参阅 *dde*(1) ）。由于命令行中没有 **-o** 选项，输出文件名是 **a.out** 。

```
ld /usr/ccs/lib/crt0.o ftn.o -lcl -lisamstub \
   -lc /opt/langtools/lib/end.o
```

在 64 位模式下执行该操作：

```
ld ftn.o -lcl -lisamstub \
   -lc /opt/langtools/lib/pa20_64/end.o
```

在 64 位模式下，链接标准模式中的共享绑定程序。请注意，没有指定 **crt0.o** ，因为对于共享链接它不再是必需的。

```
ld himom.o +std -lc
```

在 64 位模式下，链接兼容模式程序。由于 **crt0.o** 是归档链接，它已包括在其中。

```
ld /opt/langtools/lib/pa20_64/crt0.o himom.o +compat -a archive -lc
```

创建共享库：

```
ld -b -o libfunc.sl func1.o func2.o func3.o
```

创建带有内部名称的共享库：

```
ld -b -o libfoo1.1 foo1.o foo2.o +h libfoo1.1
ln -s libfoo1.1 libfoo1.sl
cc -g mytest.c -L. -lfoo1
chatr a.out
```

```
...
 共享库列表:
  dynamic   ./libfoo1.1
```

如果不使用 **+h** ，共享库将没有内部名称。链接程序不会检查 **.sl** 是否是符号链接。如果共享库没有内部名称，则将记录所查看的库名。

```
chatr a.out
...
 共享库列表:
  dynamic   ./libfoo1.sl
```

在 32 位模式下，用 **libfunc.sl** 链接程序，但使用 C 库的归档版本。在指定非致命修饰符的同时指定立即绑定模式，并允许显示详细诊断消息：

```
ld /usr/ccs/lib/crt0.o -B immediate -B nonfatal -B verbose \
   program.o -L . -lfunc -a archive -lc
```

在 64 位模式下执行该操作：

```
ld -B immediate -B nonfatal -B verbose \
   program.o -L . -lfunc -a archive -lc
```

在 32 位模式下，链接 Pascal 程序：

```
ld /usr/ccs/lib/crt0.o main.o -lcl -lm -lc
```

请注意，在以上示例中， **/usr/ccs/lib/crt0.o** 可替换为 **/opt/langtools/lib/crt0.o** 。

警告

ld 认为某些名称具有特殊的含义。符号 **_end** 由链接程序保留，特指程序地址空间结尾之外的第一个地址。同样，符号 **_edata** 特指初始化数据之外的第一个地址，符号 **_etext** 特指程序文本之外的第一个地址。符号 **end** 、 **edata** 和 **etext** 也由链接程序定义，但只有当程序包含对这些符号的引用并未定义这些符号时才如此定义（有关详细信息，请参阅 *end*(3C) ）。在 32 位模式下，符号 **__tdsize** 是程序或共享库所需的线程本地存储空间的总大小。

在 64 位模式下，链接程序会多定义几个符号。符号 **__TLS_SIZE** 是线程本地存储空间的总大小。符号 **_FPU_STATUS** 是 FPU 状态寄存器的初始状态。符号 **__SYSTEM_ID** 是任何编译单元使用的最大体系结构修订级别。

链接程序将此处列出的任何符号的用户定义当作错误。

通过其选项，链接编辑器为用户提供了很大的灵活性。但是，直接调用链接程序的用户必须承担某些附加的责任。输入选项应该确保程序的以下属性：

- 当链接编辑器通过 *cc*(1) 调用时，启动例程将与用户的程序链接。该例程在执行主程序后调用 *exit*(2)。如果用户直接调用 **ld** ，则必须确保程序始终调用 **exit()** ，而不是直通到入口例程的末尾。
- 当为与符号调试程序 *dde* 一起使用而进行链接时，用户必须确保程序包含名为 **main** 的例程。此外，用户还必须链接 **/opt/langtools/lib/end.o** （32 位）和 **/opt/langtools/lib/pa20_64/end.o** （64 位）作为命令行上命名的最后一个文件。

无法保证链接程序将按照文件在库中的相同相对顺序从归档库中提取文件并将其包含在最终程序中。

一旦检测到任何兼容性问题，链接程序将发出警告。除此之外，其他问题包括体系结构问题以及可能在一段时间内更改的功能。其中一些问题包括：

- 链接 PA 2.0 对象文件，它将不在 PA 1.x 系统上运行。
- 通过 **-A** 选项进行递增加载。
- 过程调用参数和返回类型检查，包括 **-C** 选项。
- 同名但不同类型的符号，如 CODE 和 DATA。
- 链接程序对不满足符号的检查，它有时会跳过归档库中的某些对象文件。只有在同时提供 **-v** 选项时，才会发出该警告。
- 对象在共享库中的版本控制。

这些消息可以使用 **+vnocompatwarnings** 选项关闭。

如 选项部分所述，某些选项不再存在于 64 位链接程序中。它们是：

- **-q**
- **-A**
- **-C**
- **-E**
- **-Q**
- **-S**
- **-T**
- **-X**
- **+dpv**

使用共享库和 S 位集构建的可执行文件的 64 位模式（通过 **+s** enabled).）存在某些限制。64 位可执行文件的行为不匹配对等 32 位可执行文件或用 **+compat** 构建的可执行文件的行为。对于任何 setuid 或 setgid 程序， **dld** 将禁用通过环境变量 **SHLIB_PATH** （32 位和 64 位模式）和 **LD_LIBRARY_PATH** （64 位模式）进行的任何动态库搜索。如果库仅存在于 **SHLIB_PATH** （或 **LD_LIBRARY_PATH** ）中指定的目录中，并且程序是 setuid 程序（即 uid 不等于 euid 或 gid 不等于 egid）且依赖于该库，则将收到“未找到库”运行时错误。只有当满足以下条件时， **dld** 才会使用动态路径查找（通过 **SHLIB_PATH** ）： **getuid () == geteuid () && getgid () == getegid ()** 。也就是说，如果 uid 或 gid 不匹配其有效对等项， **dld** 仅搜索记录的库路径。因此， dld 不会检查导致运行时错误“未找到库”的 **SHLIB_PATH** 。

作者

ld 由 AT&T、加州大学伯克利分校和 HP 联合开发。

文件

/usr/lib/lib*	32 位系统归档和共享库
/usr/lib/pa20_64/lib*	64 位系统归档和共享库
/usr/ccs/lib*	32 位开发归档和共享库
/opt/langtools/lib/pa20_64	64 位开发对象文件、归档和共享库
a.out	输出文件
/usr/lib/dld.sl	32 位动态加载程序
/usr/lib/pa20_64/dld.sl	64 位动态加载程序
/opt/langtools/lib/end.o	与 32 位 **dde** 调试程序一起使用
/opt/langtools/lib/pa20_64/end.o	与 64 位 **dde** 调试程序一起使用
/usr/ccs/lib/crt0.o	32 位运行时启动文件
/opt/langtools/lib/crt0.o	等效于 **/usr/ccs/lib/crt0.o**
/opt/langtools/lib/pa20_64/crt0.o	64 位运行时启动文件
/usr/ccs/lib/dyncall.o	仅限 32 位。与 **-A** 选项一起使用
/opt/langtools/lib/mcrt0.o	带配置处理的 32 位运行时启动（请参阅 *prof*(1)）
/usr/lib/milli.a	**ld** 自动搜索的 32 位 millicode 库
/usr/lib/pa20_64/milli.a	**ld** 自动搜索的 64 位 millicode 库
/opt/langtools/lib/gcrt0.o	带配置处理的运行时启动（请参阅 *gprof*(1)）
/opt/langtools/lib/icrt0.o	带配置处理的 32 位运行时启动（请参阅上面有关基于配置文件的优化的讨论）。
/usr/lib/nls/$LANG/ld.cat	消息清单
/var/tmp/ld*	临时文件
flow.data	包含通过运行提供的可执行文件生成的配置文件数据的文件
/usr/ccs/bin/fdp	用于从用提供的可执行文件创建的配置文件数据库文件创建过程链接顺序的程序；由 **-P** 选项进行派生处理
/opt/langtools/lbin/ucomp	PA-RISC 代码生成器

另请参阅

配置处理和调试工具

adb(1)	绝对调试程序
gprof(1)	显示调用图形配置文件数据
prof(1)	显示配置文件数据
dde(1)	C、C++、FORTRAN 和 Pascal 符号调试程序

系统工具

aCC(1)	调用 HP-UX aC++ 编译程序
ar(1)	创建归档库
CC(1)	调用 HP-UX C++ 编译程序
cc(1)	调用 HP-UX C 编译程序
chatr(1)	更改程序的内部属性
exec(2)	执行文件
f77(1)	调用 HP-UX FORTRAN 77 编译程序
f90(1)	调用 HP-UX Fortran 90 编译程序
fastbind(1)	调用 fastbind 工具
nm(1)	输出对象文件的名称列表
pc(1)	调用 HP-UX Pascal 编译程序
strip(1)	从对象文件删除符号和行编号信息

其他信息

a.out(4)	汇编程序、编译程序和链接程序输出
ar(4)	归档文件格式
crt0(3)	执行启动例程
dld.sl(5)	动态加载程序
end(3C)	程序中最后位置的符号

文本和教程

《HP-UX Linker and Libraries Online User Guide》
(请参阅 **+help** 选项)

《HP-UX Linker and Libraries User's Guide》
(有关订购信息，请参阅 *manuals*(5))

符合的标准

ld：SVID2、SVID3、XPG2、XPG4

名称

leave - 在需要退出时发出提醒信息

概要

leave [*hhmm*]

说明

leave 命令将等待，直到到达指定的时间，然后提醒您退出。该命令在指定时间之前的 5 分钟和 1 分钟、在该指定时间、以及指定时间后的每分钟都会提醒您退出。当您注销后 **leave** 将会退出。

退出时间采用 *hhmm* 的形式，其中 *hh* 表示小时（范围为 0 到 11 或 0 到 24）， *mm* 表示指定小时后的分钟数。如果 *hh* 的值大于 11（24 小时制），则指定的值会被减去 12，成为一个介于 0 到 11 之间的新值，从而确保报警时间始终设置为在未来 12 小时内激活。例如，如果 *hhmm* 为 1350，当前时间是 4:00 PM (1600)，则值 1350 会被更改为 150，使得报警时间设置为 1:50 AM，即 9 小时 50 分钟之后。另一个例子，如果当前时间为 9:00 AM 而 *hhmm* 指定为 2200 (10:00 PM)，则该值将被转换为 1000，使报警时间设置为 1 小时而不是指定的 13 小时之后。

如果未指定供任何参数， **leave** 将提示如下信息:

When do you have to leave?

直接回车将使 **leave** 退出；否则回复内容将被视为一个时间。这一信息适于包括在 **.login** 或 **.profile** 文件中。

leave 命令忽略中断、退出和终止信号。要关闭该命令，您应该注销或者使用 **kill -9** 命令并指定其进程 ID。

举例

以下命令:

leave 1204

将报警（嘟嘟声）发送到您的终端，提醒您必须在 12:04 退出，过了 12:04 后，将每隔一分钟提醒您一次退出时间已过。

警告

leave 命令每 100 秒钟检查一次 **/etc/utmp** 文件，了解用户是否已经注销。如果用户在 **leave** 进行定期检查之前注销后又重新登录到同一 tty，那么 **leave** 可能不会知道该用户曾注销过。

作者

leave 由加州大学伯克利分校开发。

文件

/etc/utmp

另请参阅

calendar(1)。

名称

lifcp - 复制到 LIF 文件或从 LIF 文件复制

概要

lifcp [**-T** *xxx*] [**-L** *xxx*] [**-v** *xxx*] [**-a**] [**-b**] [**-i** *xxx*] [**-r**] [**-t**] *file1 file2*

lifcp [**-T** *xxx*] [**-L** *xxx*] [**-v** *xxx*] [**-a**] [**-b**] [**-i** *xxx*] [**-r**] [**-t**] [*file1 file2* ...] *directory*

说明

lifcp 可将 LIF 文件复制到 HP-UX 文件，将 HP-UX 文件复制到 LIF 文件，或者将 LIF 文件复制到另一个 LIF 文件。它还可将 (HP-UX/LIF) 文件列表复制到 (LIF/HP-UX) 目录。参数列表上最后的名称是目标文件或目标目录。

选项

选项和参数之间的空格是可选的。

-T *xxx* 仅在将文件复制到 LIF 卷时使用。该选项强制将 LIF 目录条目的文件类型设置为指定参数。参数可以是十进制、八进制或十六进制，使用标准 C 表示法。

-L *xxx* 仅在将文件复制到 LIF 卷时使用。该选项将“最后的卷标”设置为 *xxx* （0 或 1）。缺省的“最后的卷标”为 **1** 。

-v *xxx* 仅在将文件复制到 LIF 卷时使用。该选项将“卷编号”设置为 *xxx* 。缺省的“卷编号”为 1。

-a 该选项不管文件类型，强制使用 ASCII 复制模式。以 ASCII 模式从 HP-UX 复制到 LIF 时，缺省的文件类型为 BINARY (1)。（有关可用的复制模式的详细信息，请参考 *lif*(4) ）。该选项在从 LIF 复制到 LIF 时无效。

-b 该选项不管文件类型，强制使用 BINARY 复制模式。以 BINARY 模式从 HP-UX 复制到 LIF 时，缺省的文件类型为 BINARY (-2)。（有关可用的复制模式的详细信息，请参考 *lif*(4) ）。该选项在从 LIF 复制到 LIF 时无效。

-i *xxx* 仅在将文件复制到 LIF 卷时使用。该选项将 LIF 目录条目的“implementation”字段设置为指定的参数。参数值可以是十进制、八进制或十六进制，使用标准 C 表示法。“implementation”字段只可以设置文件类型 -2001 到 -100000（八进制）。对所有交换文件类型和 -2 到 -200（八进制）文件类型，将“implementation”字段设置为 0。请注意，“implementation”值控制 LIF 文件关于保护和记录大小的属性。 **lifls -l** 或 **lifls -i** 可用于确定文件的“implementation”值。

-r 不管文件类型，强制使用 RAW 模式进行复制。以 RAW 模式从 HP-UX 复制到 LIF 时，缺省的文件类型为 BIN (-23951)。 **-T** 选项可覆盖缺省的文件类型。（各种复制模式在 *lif*(4) 中进行了说明） **-r** 选项在 LIF 到 LIF 复制操作中无效。

-t 使 HP-UX 文件名转换为 LIF 实用程序可接受的名称；也就是说，所有小写字母都转换为大写字母，并且所有其他字符（除数字以外）都更改为下划线 (_) 。如果 HP-UX 文件名不是以字母开头，则文件名前面加上大写字母 **X** 。因此，例如，如果复制了两个命名为冒号 (**:**) 和分号 (**;**) 的文件，那么这两个文件都将转换为 **X_** 。文件名被截断为最多 10 个字符。将 LIF 文

件复制到 HP-UX 或 LIF 文件时， **-t** 无效。如果使用了错误的名称，则忽略 **-t** 会导致产生错误。

下表汇总了从 LIF 复制到 HP-UX 时缺省的复制模式：

文件类型	缺省的复制模式
ASCII	ASCII
BINARY	BINARY
BIN	RAW
其他	RAW

从 HP-UX 复制到 LIF 时，缺省的复制模式为 ASCII，并且将创建一个 ASCII 文件。

从 LIF 复制到 LIF 时，如果没有指定选项，则所有 LIF 目录字段和文件内容都会从源复制到目标。

LIF 文件名由嵌入的冒号 (**:**) 分隔符来识别（有关 LIF 文件命名约定，请参阅 *lif*(4)）。 LIF 目录由结尾冒号来识别。如果使用了包含冒号的 HP-UX 文件名，则必须使用两个反斜杠字符 (****) 来转义该冒号（Shell 将删除其中一个）。

文件名 **-** （短线）被解释为表示标准输入或标准输出，这取决于它在参数列表中的位置。如果数据需要非标准转换，这将尤其有用。从标准输入进行复制时，如果没有发现其他名称，则使用名称“STDIN”。

LIF 文件命名约定仅适用于 LIF 实用程序。因为由 Shell 进行文件名扩展，所以该机制无法用于扩展 LIF 文件名。

请勿在使用 **lifcp** 时挂接特殊文件"。

诊断信息

lifcp 如果复制文件成功，则返回退出代码 0。否则将输出诊断信息并返回非零值。

举例

在 LIF 卷 **lifvol** 上，将 HP-UX 文件 **abc** 复制到 LIF 文件 **CDE** ，实际上，后者是被初始化为 LIF 卷的 HP-UX 文件：

lifcp abc lifvol:CDE

将当前目录中的所有 HP-UX 文件复制到位于父目录中的 LIF 卷 **lifvol** 。文件名转换为相应的 LIF 文件名。

lifcp -t * ../lifvol:

将当前目录中的所有 HP-UX 对象文件复制到 LIF 卷 *lifvol* 。复制模式为 RAW，并且 LIF 文件类型设置为 -5555。

lifcp -r -T -5555 -t *.o lifvol:

将当前目录中的所有对象文件复制到 LIF 卷 **lifvol** 。复制模式为 BINARY，并且创建 LIF BINARY 文件。

lifcp -r -T 0xffffe961 -i 0x20200080 bdat lifvol:BDAT

不用口令，将 BDAT 文件从 BASIC 工作站复制到 HP-UX LIF 卷 **lifvol** 。请注意， **-i** 控制保护和记录大小属性。BDAT 文件的文件类型为 -5791（或 0xffffe961），其记录大小为每记录 256 字节。

lifcp -b *.o lifvol:

将当前目录中的所有文件复制到 **root** 目录中的 LIF 卷 **lifvol** 。复制模式为 RAW，并且 LIF 文件类型设置为 BIN。

lifcp -r -t * /lifvol:

将文件 **abc:** 复制到 **lifvol** 中的 LIF 文件 **CDE** 。

lifcp abc\\: lifvol:CDE

将文件 **abc** 和 **def** 复制到 **lifvol** 中的 LIF 文件 **ABC** 和 **DEF** 。

lifcp -t abc def lifvol:

将 **lifvol** 中的 LIF 文件 **ABC** 复制到当前目录中的文件 **ABC** 。

lifcp lifvol:ABC .

将标准输入复制到 LIF 卷 **/dev/dsk/c0t6d0** 中的 LIF 文件 **A_FILE** 。

lifcp - /dev/dsk/c0t6d0:A_FILE

将 **lifvol** 中的 LIF 文件 **ABC** 复制到 **/dev/dsk/c0t6d0** 中的 LIF 文件 **CDE** 。

lifcp lifvol:ABC /dev/dsk/c0t6d0:CDE

将 *pr* 的输出复制到 LIF 文件 **ABC** 。

pr abc | lifcp - lifvol:ABC

将 *pr* 的输出复制到 LIF 卷 **lifvol** 。由于没有指定文件名，因此将创建 LIF 文件 **STDIN** 。

pr abc | lifcp - lifvol:

将 **lifvol** 中的 LIF 文件 **ABC** 复制到标准输出。

lifcp lifvol:ABC -

将当前目录中的所有文件复制到 LIF 卷 **lifvol** 中同名的 LIF 文件（如果当前目录中的文件名不符合 LIF 命名约定，则可能导致错误！）。

lifcp * ../lifvol:

相关内容

700/800 系列

还支持下列选项：

-K*nnn* 强制在其中复制的每个文件都从卷的开始以 *nnn* ?1024 字节边界开始。在将文件用于 700/800 系列引导介质时非常有用。该选项在从 LIF 卷复制时无效。

作者

lifcp 由 Hewlett-Packard Company 开发。

另请参阅

lifinit(1)、 lifls(1)、 lifrename(1)、 lifrm(1)、 lif(4)。

名称

lifinit - 在文件上写入 LIF 卷标题

概要

lifinit [**-v***nnn*] [**-d***nnn*] [**-n** *string*] [**-s***nnn*] [**-l***nnn*] [**-e***nnn*] *file*

说明

lifinit 可在卷或文件上写入 LIF 卷标题。

选项

lifinit 可识别下列选项和命令行参数，这些选项和参数可以按任何顺序排列：

-v*nnn* 将卷大小设置为 *nnn* 字节。如果 *nnn* 不是 256 的倍数，则将其向下舍入到最接近的倍数值。

-d*nnn* 将目录大小设置为 *nnn* 个文件条目。如果 *nnn* 不是 8 的整数倍数，则将其向上舍入到最接近的整数倍数值。

-n *string* 将卷名设置为 *string* 。如果未指定 **-n** 选项，则卷名将设置为由 *file* 指定的路径名的最后组成部分。合法的 LIF 卷名长度为 6 个字符，并且被限制为大写字母 (A-Z)、数字 (0-9) 和下划线字符 (_)。第一个字符（如果有）必须是一个字母。实用程序将自动执行转换，以创建合法的 LIF 卷名。因此，所有小写字母将被转换为大写字母，并且所有其他字符（数字和下划线除外）将被替换为大写字母 **X** 。如果卷名不是以字母开头，则在卷名之前将添加大写字母 **X**。在需要时，也会以空格填补卷名的右部或者截断卷名，使其长度达到六个字符。如果使用 **-n** 时不带 *string* ，则缺省的卷名将设置为 6 个空格。

-s*nnn* 在卷标中将初始系统加载 (ISL) 的起始地址设置为 *nnn* 。这在构建 700/800 系列系统的引导介质时非常有用。

-l*nnn* 指定 LIF 卷中 ISL 代码的长度（以字节为单位）。

-e*nnn* 从 ISL 的开头起将 ISL 入口点设置为 *nnn* 字节。例如，指定 **-e3272** 表示从 ISL 对象模块的开头起， ISL 入口点为 3272（十进制）字节。

-K*nnn* 强制目录的起始位置为从卷的开头起与 *nnn* × 1024 字节倍数最接近的地方。这对于从 LIF 介质引导 700/800 系列系统是非常必要的。

如果 *file* 不存在，则将创建并初始化常规的 HP-UX 磁盘文件。

对于常规文件来说，卷大小的缺省值为 256 千字节，对于设备文件来说，该值是设备的实际容量。

缺省目录大小是卷大小的函数。对于卷目录而言，所分配的卷大小的百分比如下：

卷大小	目录大小
< 2MB	˜1.3%
> 2MB	˜0.5%

每个目录条目占用存储器的 32 字节。实际的目录空间将受到上述舍入规则的影响。

在使用 **lifinit** 时， 请勿挂接特殊文件。

返回值

如果成功初始化卷，则 **lifinit** 返回退出代码 0。否则，将输出诊断消息并返回非零值。

举例

初始化文件 **x** ，使其成为 LIF 卷，其中包含 500 000 字节，并含有 10 个目录文件条目：

lifinit -v500000 -d10 x

使用缺省的初始化条件（设备不得是已安装的文件系统设备）来初始化设备 **/dev/rdsk/c0t6d0** ，使其成为 LIF 卷：

lifinit /dev/rdsk/c0t6d0

警告

为避免损坏介质，一旦开始执行 *lifinit* ，就不要终止它。

作者

lifinit 由 HP 开发。

另请参阅

lifcp(1)、 lifls(1)、 lifrename(1)、 lifrm(1)、 lif(4)。

名称

lifls - 列出 LIF 目录的内容

概要

lifls [**option**] *name*

说明

lifls 可在标准输出中列出 LIF 目录的内容。如果标准输出为字符专用文件，则缺省输出格式为以多列形式列出文件名（类似于 *ls*(1)，但未排序）。如果标准输出不是 tty 设备，则输出格式为每行一个文件名。 *name* 表示 HP-UX 文件的路径名，由 LIF 卷和可选的文件名组成。如果 *name* 是卷名，则列出整个卷。如果 *name* 是 *volume***:***file* 的形式，则仅列出该文件。该命令可以使用下列选项，且对于一个给定的命令只能使用一个选项:

-l 以长格式列出，给出卷名、卷大小、目录起始位置、目录大小、文件类型、文件大小、文件起始位置、“implementation”字段（十六进制）、创建日期、结束卷和卷号。

-C 不管标准输出类型如何，强制使用多列输出格式。

-L 以十进制返回“last volume flag”的内容。

-i 以十六进制返回“implementation”字段的内容。

-v 以十进制返回“volume number”的内容。

-b *blist* 仅报告使用命令行上 *blist* 中指定的块编号的文件， *blist* 是由逗号分隔的块（以 DEV_BSIZE 为单位）编号列表。

在使用 **lifls** 时请勿挂接专用文件。

诊断信息

lifls 在成功列出目录时返回零。否则将输出诊断信息并返回非零值。

举例

lifls -C /dev/rdsk/c0t6d0

作者

lifls 由 HP 开发。

另请参阅

lifcp(1)、 lifinit(1)、 lifrename(1)、 lifrm(1)、 lif(4)。

名称

lifrename - 重命名 LIF 文件

概要

lifrename *oldfile newfile*

说明

oldfile 是要重命名的文件（如 **liffile:A_FILE** ）的完全 LIF 文件说明符（有关详细信息，请参阅 *lif*(4) ）。 *newfile* 是要指定给该文件的新名称（仅限文件名部分）。这一操作不包括复制或删除。即使将重命名的文件名不是合法的 LIF 名称，旧文件名也必须与该文件名相匹配。

请勿在使用 **lifrename** 时挂接专用文件。

诊断信息

lifrename 如果成功更改了文件名，则返回零。否则将输出诊断信息并返回非零值。

举例

lifrename liffile:A_FILE B_FILE

lifrename /dev/dsk/c0t6d0:ABC CDE

作者

lifrename 由 HP 开发。

另请参阅

lifcp(1)、 lifinit(1)、 lifls(1)、 lifrm(1)、 lif(4)。

名称

lifrm - 删除 LIF 文件

概要

lifrm *file1* ... *filen*

说明

lifrm 从 LIF 卷中删除一个或多个条目。文件名说明符如 *lif*(4) 中所述。

请勿在使用 **lifrm** 时挂接特殊文件"。

诊断信息

lifrm 如果成功删除了文件，则返回零。否则将输出诊断信息并返回非零值。

举例

lifrm liffile:MAN

lifrm /dev/rdsk/c0t6d0:F

作者

lifrm 由 HP 开发。

另请参阅

lifcp(1)、lifinit(1)、lifls(1)、lifrename(1)、lif(4)。

名称

line - 从用户输入中读取一行

概要

line [**-t** *timeout*]

说明

line 从标准输入中复制一行（直到换行符为止），并将该行写入到标准输出中。遇到 EOF 时，该命令将返回退出代码 1，并始终至少输出一个换行符。该命令通常在 Shell 文件中用于从用户终端读取数据。

选项

line 可识别以下命令行选项:

-t *timeout* 在 *timeout* 指定的秒后发生超时，此时 *timeout* 是一个整数值（如果指定的是非整数值，则会被转换为整数；即进行四舍五入）。在 **-t** 和 *timeout* 参数之间需要留一个空格。该选项未记录在 POSIX 及其他行业标准中，因此不应该用于可移植应用程序中。

外部语言环境影响

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

以下 Shell 脚本行可提示输入一个文件名并显示有关该文件的信息:

```
echo 'Enter file name: \c'
reply=`line`
ls -l $reply
```

要将响应时间限制为 10 秒，可使用以下格式:

```
reply=`line -t 10`
```

然后测试看有没有响应。如果在超时时间到期之前没有发生响应，则应该提供一个缺省操作。

警告

此命令可能会从 X/Open 标准中取消。使用该命令的应用程序对于其他供应商的系统来说可能是不可移植的。建议使用 **read** 作为替代命令。

另请参阅

sh(1)、 read(2)。

符合的标准

line：SVID2、SVID3、XPG2、XPG3

名称

listusers - 显示用户登录数据

概要

listusers [**-g** *groups*] [**-l** *logins*]

说明

listusers 命令显示与用户登录有关的数据。输出结果显示用户登录名和 **/etc/passwd** 注释字段值（如用户名等）。缺省情况下，显示所有用户登录的数据。

选项

listusers 命令支持下列选项:

-g *groups* 显示属于 *groups* 的所有用户，按登录的先后顺序排序。可使用由逗号分隔的列表指定多个组。

-l *logins* 显示请求的 *logins* 。可使用由逗号分隔的列表指定多个登录。

一个用户登录的 UID 为 100 或更大。

当同时使用 **-l** 和 **-g** 选项时，一个用户登录将只显示一次，即使该登录属于多个指定的组亦是如此。

举例

列出所有用户登录。

listusers

列出组 **cmds** 中的所有用户登录及用户 **bob**、**john** 和 **otto**，删除所有重复的用户登录。

listusers -g cmds -l bob,john,otto

文件

/etc/passwd

/etc/group

另请参阅

passwd(1)、 logins(1M)、 group(4)、 passwd(4)。

符合的标准

listusers： SVID3

名称

ln - 链接文件和目录

概要

ln [**-f**] [**-i**] [**-s**] *file1 new_file*

ln [**-f**] [**-i**] [**-s**] *file1* [*file2* ...] *dest_directory*

ln [**-f**] [**-i**] [**-s**] *directory1* [*directory2* ...] *dest_directory*

说明

ln 命令的作用如下:

- 将 *file1* 链接到新的或现有的 *new_file* ，
- 将 *file1* 链接到现有 *dest_directory* 中名为 *file1* 的新文件或现有文件，
- 将 *file1* 、 *file2* ... 链接到现有 *dest_directory* 中同名的新文件或现有文件，
- 将 *directory1* 、 *directory2* ... 链接到现有 *dest_directory* 中同名的新目录，
- 或者，它在文件之间或目录之间创建符号链接。

如果链接指向 *dest_directory* ，则该目录中的对应文件或目录名称将相应地链接到 *file1* 、 *file2* ... 或 *directory1* 、 *directory2* ... 等。如果指定了两个或更多现有文件或目录（不包括目标文件名 *new_file* ），则目标必须是目录。如果 *new_file* 已作为常规文件（或指向其他文件的链接）存在，则仅当指定 **-f** 选项时才删除其内容（或现有链接）及其 ACL。链接后的 *new_file* 上的 ACL 与 *source_file* 文件上的相同。

如果指定了 **-f** 和 **-i** 选项，而且要创建的链接是现有链接或普通文件的名称，文件的访问权限禁止写入，则 **ln** 要求覆盖文件的权限。如果目录的访问权限禁止写入，则 **ln** 将异常中止并返回以下错误消息:

> **cannot unlink** *new_file*

（即使文件为普通文件而且不是指向其他文件的链接）。当要求覆盖现有文件或链接的权限时， **ln** 将输出模式（请参阅下面的 *chmod*(2) 和访问控制列表），该模式后跟当前本地语言中单词 **yes** 和 **no** 的首字母，提示您作出响应，并从标准输入读取一行。如果响应是肯定的、可允许的，则将发生该操作；如果不是这样，则命令将继续到下一个源文件（如果有的话）。

硬链接是使用与它们所链接到的文件或目录相同的所有权和权限创建的。如果在链接或文件上更改了所有权或权限，则在对应的硬链接上将出现相同的更改。 **ln** 命令不允许指向目录的硬链接。

符号链接是使用创建者的所有权创建的，该权限是创建者当前的 umask。在创建后，符号链接的所有权和权限不会发生改变，因为系统忽略符号链接的模式和所有权。

如果 *file1* 是文件，而 *new_file* 是指向现有文件的链接或具有其他链接的现有文件，则取消 *new_file* 与现有文件和链接的关联，并将它链接到 *file1* 。当 **ln** 创建指向新文件名或现有文件名的链接时，所有权和权限始终与它所链接到的文件的相同。如果使用 **chown** 、 **chgrp** 或 **chmod** 更改文件或链接的所有权或权限，则更改将应用于该文件和所有关联的链接。文件和所有关联链接的上次修改时间和上次访问时间是相同的（请参阅 *chown*(1) 和 *chmod*(1) ）。

有关符号链接的讨论，请参阅 *symlink*(4) 。

选项

ln 命令可识别下列选项:

- **-f** 强制删除现有目标路径名以允许链接。
- **-i** 将提示写入标准错误输出，要求对将覆盖现有文件的每个链接进行确认。此选项仅当与 **-f** 选项联合使用时才生效。
- **-s** 导致 **ln** 创建符号链接而不是通常的硬链接。符号链接包含它所链接到的文件的名称。对链接执行 **open()** 操作时，使用引用的文件（请参阅 *open*(2) ）。符号链接上的 **stat()** 返回链接到的文件；必须执行 **lstat()** 才能获得有关链接的信息（请参阅 *stat*(2) ）。可以使用 **readlink()** 调用读取符号链接的内容（请参阅 *readlink*(2) ）。符号链接可能跨文件系统和引用目录。

访问控制列表 (ACL)

如果可选的 ACL 条目与 *new_file* 关联，则在要求覆盖文件的权限时， **ln** 会在访问模式后面显示一个加号 (**+**) 。

如果 *new_file* 是一个新文件，则它继承 *file1* 的访问控制列表，并进行更改以反映这两个文件之间的任何所有权差异（请参阅 *acl*(5) 和 *aclv*(5) ）。在 JFS 文件系统中，由 **ln** 创建的新文件不继承其父目录的缺省 ACL 条目（如果有的话），而是保留其原始 ACL。

外部语言环境影响

环境变量

LC_CTYPE 用于确定将文本解释为单字节和（或）多字节字符的方法。

LANG 和 **LC_CTYPE** 用于确定 **y** 的等效本地语言（用于是/否查询）。

LANG 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值 **C** （请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **ln** 就会认为所有国际化变量都设置为 **C** 。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

以下命令在 **dest_dir** 中创建 **file1** 和 **file2** （它们链接回原始文件 **file1** 和 **file2** ）:

ln -f file1 file2 dest_dir

如果 **file1** 和（或） **file2** 存在于目标目录中，则将删除它，并将其分别替换为指向 **file1** 或 **file2** 的链接。如果现有文件 **file1** 或 **file2** 是指向其他文件的链接或具有链接的文件，则将保留现有文件。只有链接被中断并替换为指向 **file1** 或 **file2** 的新链接。

警告

ln 不会跨文件系统创建硬链接。

相关内容

NFS

网络文件的访问控制列表将被汇总（如通过 **stat()** 在 **st_mode** 中返回的值），但不会复制到新文件中。在这样的文件上使用 **ln** 时，如果要求覆盖文件的权限，则不在模式值的后面输出 **+** 。

作者

ln 由AT&T、加州大学伯克利分校和 HP 联合开发。

另请参阅

cp(1)、 cpio(1)、 mv(1)、 rm(1)、 link(1M)、 readlink(2)、 stat(2)、 symlink(2)、 symlink(4)、 acl(5)、 aclv(5)。

符合的标准

ln：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

locale - 获取语言环境特有 (NLS) 信息

概要

locale [**-a** [**32** | **64**] | **-A** | **-m**]

locale [**-ck**] *name* ...

locale [**-pa32**] [**-pa64**]

说明

locale 命令显示有关当前语言环境或可用语言环境的信息。

在不使用参数调用时， **locale** 按如下所示的顺序显示每个语言环境相关环境变量的名称及实际值或隐含值，每行显示一个：

LANG
LC_CTYPE
LC_COLLATE
LC_MONETARY
LC_NUMERIC
LC_TIME
LC_MESSAGES
LC_ALL

实际值是变量在用户环境中实际具有的值。隐含值派生自其他变量的值。显示的隐含值括在双引号中，而实际值不括在引号中。

隐含值是这样确定的：如果变量 **LC_ALL** 存在且具有非空值，则该值为 **LC_ALL** 的实际值，并且所有其他变量都采用该值作为隐含值。如果未设置 **LC_ALL** ，则显示设置的所有 **LC_*** 变量时将其值作为实际值。显示没有值的任何变量时会将 **LANG** 环境变量的值作为其隐含值。如果 **LC_ALL** 没有值，则将它显示为 **LC_ALL=\n** 。

locale 命令可以采用多个参数，其参数可以是语言环境类别名称、语言环境关键字或特殊字 **charmap** （有关语言环境关键字和字符映射的说明，请参阅 *localedef*(1M) ）。如果参数是关键字，则显示当前环境中与该关键字关联的值，还可能显示其他信息，具体取决于所选的选项。如果参数是类别名称（即 **LC_*** ），则显示在该类别中定义的所有关键字的值。如果参数是特殊字 **charmap** ，则将显示在当前语言环境定义中使用的字符映射文件（如果有的话）。

不可输出的字符按以下格式作为十六进制值打印：

\x*hh*

例外情况是，如果已经为语言环境定义了不同的转义符，则将显示它而不是“\”。

选项

可以使用以下选项：

-a 列出所有的可用语言环境。它们是可以分配给 **LANG** 或系统上任何 **LC_*** 变量的可能的有意义值。它们取决于在系统上安装的语言环境。缺省情况下，在 PA-RISC 系统上，列出 **/usr/lib/nls/loc/locales** 中的语言环境。缺省情况下，在基于 Itanium(R) 的系统上，列出 **/usr/lib/nls/loc/hpux32/locales** 中的语言环境。此选项采用 32（对于 ILP32，为 32 位 int、long、pointer、32 位偏移量）或 64（对于 LP64，为 64 位 long、pointer、64 位偏移量）作为其参数。

- **-a** 对于 PA-RISC 和基于 Itanium 的系统，显示 32 位语言环境。
- **-a 32** 对于 PA-RISC 和基于 Itanium 的系统，显示 32 位语言环境。
- **-a 64** 在 64 位系统上仅显示 64 位语言环境。如果在 32 位系统上执行，则将返回错误消息。

-A 列出系统上可用的基于 Itanium 的系统 32 位语言环境、基于 Itanium 的系统 64 位语言环境、PA-RISC 32 位语言环境和 PA-RISC 64 位语言环境。

-m 显示系统上可用字符映射文件的列表。有关字符映射文件的定义及其用法，请参阅 *localedef*(1M)。

-c 显式地显示已经选择的语言环境类别的名称，或通过提供其中包含的关键字显示它们。此选项可以与 **-k** 选项一起使用。

-k 显式地显示已经选择的关键字的名称，或者通过以参数形式提供其包含类别显示它们。关键字的名称和值显示为：

<*keyword*>**=**<*value*>

如果没有 **-k** 选项，则仅显示值。此选项可以与 **-c** 选项一起使用。

-pa32 显示 32 位 PA-RISC 语言环境（仅在基于 Itanium 的系统上可用的选项）。

-pa64 显示 64 位 PA-RISC 语言环境（仅在基于 Itanium 的系统上可用的选项）。

name 指定语言环境类别名称、语言环境关键字或特殊字 **charmap**。

外部语言环境影响

环境变量

LANG 为没有设置或设置为 null（空）的国际化变量提供了缺省值。如果 **LANG** 未设置或设置为 null（空），则会使用缺省值“C”（请参阅 *lang*(5)）。如果任一国际化变量中包含无效设置，则 **locale** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5)。

LC_ALL，当设置为非空字符串值时，将覆盖所有其他国际化变量的值。

LC_CTYPE 用于确定文本解释为单字节和（或）多字节字符，确定字符分类为可打印字符，以及确定正则表达式中由字符类表达式匹配的字符。

LC_MESSAGES 用于确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息，以及写入到标准输出的信息消息的格式和内容。

NLSPATH 用于确定消息目录的位置，以便处理 **LC_MESSAGES** 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

locale 命令退出时返回下列值之一：

- **0** 找到了所有所需信息并成功显示。
- **>0** 在查找或显示信息时出现错误。

举例

如果将语言环境变量设置为：

LANG=fr_FR.iso88591
LC_COLLATE=C

命令：

locale

提供以下输出：

LANG=fr_FR.iso88591
LC_CTYPE="fr_FR.iso88591"
LC_COLLATE=C
LC_MONETARY="fr_FR.iso88591"
LC_NUMERIC="fr_FR.iso88591"
LC_TIME="fr_FR.iso88591"
LC_MESSAGES="fr_FR.iso88591"
LC_ALL=

命令：

LC_ALL=POSIX locale -ck decimal_point

生成：

LC_NUMERIC decimal_point="."

如果将 **LANG** 设置为 **POSIX** 且未设置其他语言环境变量，则命令：

locale LC_NUMERIC

生成：

"."
""

""

它对应于关键字 *decimal_point* 、 *thousands_sep* 、 *grouping* 和 *alt_digit* 。

另请参阅

localedef(1M) 、 localeconv(3C) 、 nl_langinfo(3C) 、 setlocale(3C) 、 charmap(4) 、 localedef(4) 、 environ(5) 、 lang(5) 。

符合的标准

locale： XPG4、POSIX.2

名称

lock - 保护终端

概要

lock

说明

lock 要求用户输入口令，然后在终端上输出 **LOCKED** ，直至重新输入口令后才放弃终端的锁定。如果用户忘记了口令，唯一的解决方法是从别处登录并终止锁定进程。

名称

logger - 生成系统日志中的条目

概要

logger [**-t** *tag*] [**-p** *pri*] [**-i**] [**-f** *file*] [*message* ...]

说明

logger 命令提供了与 **syslog()** （请参阅 *syslog*(3C) ）系统日志模块的程序接口。

可以在命令行上提供一条消息，然后直接在日志中记录；或者读取一个文件，并记录每一行。如果未指定任何 *file* 或 *message* ，则记录标准输入的内容。

选项

logger 命令可识别下列命令行选项和参数：

-t *tag* 使用指定的 *tag* 来标记日志中的每一行。缺省值是 **getlogin()** （请参阅 *getlogin*(3C) ）的返回值。如果 **getlogin()** 返回 NULL，则 **syslog** 是缺省值。

-p *pri* 输入具有指定优先级的消息。可以数字形式或 *facility*.*level* 对的形式指定优先级。例如， **-p local3.info** 将记录 **local3** 设备中具有 **info** 级别的消息。缺省值为 **user.notice** 。

-i 使用每一行记录 logger 进程的进程 ID。

-f *file* 记录指定文件的内容。

message 要记录的消息；如果未指定，则记录由 **-f** 选项指定的文件或标准输入的内容。

外部语言环境影响

环境变量

LC_MESSAGES 用于确定显示消息的语言。

如果在环境中未指定 **LC_MESSAGES** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 。

如果任一国际化变量中包含无效设置，则 **logger** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

将消息 **System rebooted** 发送到 **syslogd** 守护程序：

logger System rebooted

将 **users** 命令（请参阅 *users*(1) ）的输出发送到 **syslogd** 守护程序，并标记为 **info** 级别 **local0** 工具。使用字符串 **USERS** 标记消息：

users | logger -p local0.info -t USERS

将 **user** 设备、 **emerg** 级别的消息 **System going down immediately!!!** 发送到 **syslog** 守护程序：

logger -p user.emerg "System going down immediately!!!"

警告

如果未在系统中运行 **syslogd** 守护程序（请参阅 *syslogd*(1M)），则 **logger** 命令没有任何作用。

不支持以非 POSIX/C 语言环境输出的消息。

作者

logger 由加州大学伯克利分校开发。

另请参阅

syslogd(1M)、 getlogin(3C)、 syslog(3C)。

符合的标准

logger： XPG4、POSIX.2

名称

login - 登录；启动终端会话

概要

login [*name* [*env-var*] ...]

说明

login 命令在每个终端会话的开头用来正确地标识可能的用户。 **login** 可以作为用户命令调用，也可以在建立传入连接时由系统调用。 **login** 还可以在先前用户 Shell 终止但终端尚未断开连接时由系统调用。

如果 **login** 作为命令调用，它必须替换初始命令解释程序（用户的登录 Shell）。这可以使用以下 Shell 命令来完成

exec login

如果未在命令行上指定用户的登录名，则请求该登录名，并且通过以下提示获取相应的口令（如果是必需的）：

login:
Password:

口令输入过程中将关闭终端回显（如果可能），以防止以书面形式记录口令。如果帐户没有口令，而帐户的身份验证配置文件却需要口令， **login** 将调用 **pam_chauthtok()** 为帐户创建口令。在可信系统上， **login** 将显示上次的成功和不成功登录次数和终端设备。

作为安全预防措施，某些安装使用需要另一个“拨号”口令的选项。这仅用于拨号连接，并且使用以下提示进行请求：

dialup password:

要成功登录，这两个口令均必须正确（有关拨号安全机制的详细信息，请参阅 *dialups*(4) ）。

如果激活口令老化机制，用户的口令可能已过期。将调用 **pam_chauthtok()** 来更改口令。在非可信环境中，将在成功的口令更改后要求用户重新登录（请参阅 *passwd*(1) ）。

三次不成功的登录尝试后，将发出 **HANGUP** 信号。如果登录未在特定的时间（如一分钟）内成功完成，终端将以静默方式断开连接。

成功登录后，将更新记账文件，初始化用户与组 ID、组访问列表和工作目录，并从文件 **/etc/passwd** 和 **/etc/logingroup** 中的相应用户条目中确定用户的命令解释程序 (Shell)（请参阅 *passwd*(4) 和 *group*(4) ）。如果 **/etc/passwd** 没有为用户名指定 Shell，则将在缺省情况下使用 **/usr/bin/sh** 。然后， **login** 使用 Shell 路径名的最后一部分前加 **-** 的形式（例如 **-sh** 或 **-ksh** ）派生适当的 Shell。以这种名称前加减号的形式调用命令解释程序时， Shell 将执行其自己的初始化，包括执行配置文件、登录或其他初始化脚本。

例如，如果用户登录 Shell 是 Korn 或 POSIX Shell（请分别参阅 *ksh*(1) 或 *sh-posix*(1) ）， Shell 将执行配置文件 **/etc/profile** 和 **$HOME/.profile** （如果它们存在），并可能执行其他脚本。根据这些配置文件包含的内容，可能会显示有关用户邮件文件中邮件的消息，或者用户自上次登录以来可能收到的任何消息。

如果命令名字段是 *** ，则对该条目的目录字段中指定的目录执行 **chroot()** 。此时，将在新级别执行 **login** ，该级

别必须具有其自己的根结构（包括 **/usr/bin/login** 命令和 **/etc/passwd** 文件）。

对于普通用户，基本的环境变量（请参阅 *environ*(5) ）初始化为：

> **HOME=***login_directory*
> **LOGNAME=***login_name*
> **MAIL=/var/mail/***login_name*
> **PATH=:/usr/bin**
> **SHELL=***login_shell*

login_directory 、 *login_name* 和 *login_shell* 从 **passwd** 文件条目的相应字段中提取（请参阅 *passwd*(4) ）。

对于超级用户， **PATH** 设置为：

> **PATH=:/usr/sbin:/usr/bin:/sbin**

对于远程登录，环境变量 **TERM** 也设置为远程用户的终端类型。

通过在执行时间或当 **login** 请求用户的登录名时给 **login** 提供附加的参数，可以扩展或修改环境。参数的形式可以是 *value* 或 *varname***=***value* ，其中 *varname* 是新的或现有的环境变量， *value* 是赋给该变量的值。

第一种形式的参数（无等号）以如下输入形式放入环境中

> **L***n***=***value*

其中 *n* 是从 0 开始，在每次请求新的变量名时递增的数字。

第一种形式的参数（带等号）不做修改即放入环境中。

如果环境变量 (**L***n* 或 *varname*) 已经出现在环境中，则新值将替换旧值。

有两种例外情况。不能更改变量 **PATH** 和 **SHELL** 。这将防止以受限 Shell 环境登录的用户派生不受限制的二级 Shell。

login 和 **getty** 均理解简单的单字符引用约定。在字符前键入反斜杠将引用该字符，并允许包含空格和制表符等字符。

如果存在 **/var/adm/btmp** ，所有不成功的登录尝试记录到该文件中。如果不存在该文件，则引用该功能。 **lastb** 命令（请参阅 *last*(1) ）显示具有 **btmp** 读取访问权的用户的无效登录尝试的摘要。

如果存在 **/etc/securetty** 文件，则登录安全机制将生效，即只允许 **root** 在该文件中列出的 tty 上成功登录。受限的 tty 按设备名列出，每行列出一个。有效的 tty 名称取决于具体的安装。示例

> **console**
> **tty01**
> **ttya1**
> 等

请注意，该功能不禁止普通用户使用 **su** 命令（请参阅 *su*(1) ）。

HP-UX 智能卡登录

如果配置用户帐户使用智能卡，用户口令将存储在智能卡中。该口令与系统上存储的普通口令具有相同的特性。

要使用智能卡帐户登录，必须将智能卡插入智能卡读卡器。系统将在身份验证过程中提示用户输入 PIN （个人标识号）而不是口令。这些提示是：

login:
Enter PIN:

当输入有效的 PIN 时，将自动从智能卡中检索口令。因此，不必要知道口令，知道 PIN 即可。

如果连续三次输入错误的 PIN，则将锁定智能卡。只有发卡方才能将其解锁。

安全功能

在标准系统上，如果存在以下任一种情况， login 将禁止用户登录：

- 帐户的口令已过期，并且用户无法成功更改该口令。
- 帐户的口令已过期，并且在过期后的指定天数内没有更改口令（请参阅 shadow(4)）。
- 帐户的使用周期已结束（请参阅 shadow(4)）。

在可信系统上，如果存在以下任意情况， **login** 将禁止用户登录：

- 帐户的口令已过期，并且用户无法成功更改该口令。
- 帐户的口令使用周期已结束。
- 上次登录和当前时间之间的时间超过允许的登录间隔时间。
- 已经在帐户上设置管理锁。
- 已经超过帐户的不成功登录尝试最大次数。
- 已经超过终端的不成功登录尝试最大次数。
- 已经在终端上设置管理锁。
- 终端包含授权用户列表，而当前用户不在列表中。
- 终端包含一天中特定时间的限制，而当前时间不在允许的期限内。

在可信系统上， **login** 允许超级用户在控制台上登录，除非 **/etc/securetty** 存在且不包含 **console** 。

有关影响该命令行为的可配置参数的详细信息，请参阅 *security*(4) 联机帮助页中的 **/etc/default/security** 文件。目前支持的参数是：

ABORT_LOGIN_ON_MISSING_HOMEDIR

NOLOGIN

NUMBER_OF_LOGINS_ALLOWED

外部语言环境影响

环境变量

HOME 用户的主目录。

MAIL 查找邮件的位置。

PATH 搜索命令的路径。

SHELL 所使用的命令解释程序。

TERM 用户的终端类型。

varname 用户指定的命名变量。

L*n* 用户指定的未命名变量。

诊断信息

如果出现关联的情况，则将显示以下诊断信息：

.rhosts is a soft link

个人等同权限文件是符号链接。

Bad .rhosts ownership

个人等同权限文件不为本地用户或拥有适当特权的用户所拥有。

Bad group id

setgid() 失败（请参阅 *setuid*(2)）。

Bad user id

setuid() 失败（请参阅 *setuid*(2)）。

Cannot open password file

请向系统管理员咨询。

Locuser too long

指定的字符串对于 **login** 的内部缓冲区来说太长。

Login incorrect

用户名和口令无法匹配。

No /usr/bin/login or /etc/login on root

试图登录不包含子根登录命令的子目录根。也就是说， **passwd** 文件条目包含 Shell 路径 * ，但系统无法在给定的主目录下找到 **login** 命令。

No directory

请向系统管理员咨询。

No Root Directory

试图登录不存在的子目录根。也就是说， **passwd** 文件条目包含 Shell 路径 * ，但系统对给定的主目录执行 **chroot()** 。

No shell

用户 Shell（如果 Shell 名称在 **/etc/passwd** 中为空，则为 /usr/bin/sh ）不能以 **exec** 命令启动。请向系统管理员咨询。

No utmp entry. You must exec "login" from the lowest level "sh"

试图不使用 Shell 的 **exec** 内部命令或从其他初始 Shell 中将 **login** 作为命令来执行。当前的 Shell 已终止。

Remuser too long

指定的字符串对于 **login** 的内部缓冲区来说太长。

Terminal type too long

指定的字符串对于 **login** 的内部缓冲区来说太长。

Unable to change to directory *name*

无法通过 **chdir** 更改到用户的主目录。

Your password has expired. Choose a new one

口令老化功能已启用，而用户的口令已过时。

警告

如果 **/etc/group** 链接到 **/etc/logingroup** ，尝试登录的用户的组组成员关系由网络信息服务 (NIS) 来进行管理，并且所有 NIS 服务器均不能响应， **login** 将等到服务器响应为止。

相关内容

可插拔身份验证模块 (PAM)

PAM 是用户身份验证、口令修改和帐户验证的 Open Group 标准。尤其是， **pam_authenticate()** 将调用来执行与 **login** 相关的所有功能。其中包括检索口令，验证帐户和显示错误消息。 **pam_chauthtok()** 在口令过期或建立的过程中调用。

HP Process Resource Manager

如果安装并配置可选的 HP Process Resource Manager (PRM) 软件，登录 Shell 将在用户初始进程资源组中启用。如果未定义用户的初始组， Shell 将在用户缺省组 (**PRMID=1**) 中运行。有关如何配置 HP PRM 的说明，请参阅 *prmconfig*(1) ；有关如何确定用户初始进程资源组的说明，请参阅 *prmconf*(4) 。

作者

login 由 AT&T 和 HP 联合开发。

文件

$HOME/.profile	个人配置文件（单个用户初始化）
$HOME/.rhosts	远程登录服务器的个人等同权限文件。
/etc/d_passwd	拨号安全加密口令。
/etc/dialups	需要拨号安全机制的行。
/etc/hosts.equiv	允许不使用口令登录的等效主机的系统列表。
/etc/logingroup	组文件 — 定义组访问列表。
/etc/motd	当天的消息。
/etc/passwd	口令文件 — 定义用户、口令和主要组。
/etc/profile	系统配置文件（所有用户的初始化）。
/etc/securetty	根目录登录的有效 tty 的列表。
/etc/shadow	影子口令文件。
/etc/utmp	当前登录的用户。
/tcb/files/auth/*/*	可信系统口令数据库。
/var/adm/btmp	无效登录尝试的历史信息。
/var/adm/wtmp	登录、注销和日期更改的历史信息。
/var/mail/*login_name*	用户的邮箱。 *login_name*
/etc/default/security	安全缺省值配置文件。

另请参阅

csh(1)、 groups(1)、 ksh(1)、 last(1)、 mail(1)、 newgrp(1)、 passwd(1)、 sh(1)、 sh-posix(1)、 su(1)、 getty(1M)、 initgroups(3C)、 dialups(4)、 group(4)、 passwd(4)、 profile(4)、 security(4)、 shadow(4)、 utmp(4)、 environ(5)。

HP Process Resource Manager

《 HP Process Resource Manager Users Guide 》中的 prmconfig(1)、 prmconf(4)

可插拔身份验证模块 (PAM)

pam_acct_mgmt(3)、 pam_authenticate(3)、 pam_chauthtok(3)。

HP-UX 智能卡登录

scpin(1)、 scsync(1)。

名称

logname - 获取登录名

概要

logname

说明

logname 将用户的登录名写入标准输出。该登录名等同于 **getlogin()** 返回的登录名（请参阅 *getlogin*(3C) ）。

外部语言环境影响

环境变量

LANG 确定显示的诊断消息所用的语言。

文件

/etc/profile

另请参阅

env(1)、login(1)、getlogin(3C)、logname(3C)、environ(5)。

符合的标准

logname：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

lorder - 查找对象库的排序关系

概要

lorder [*files*]

说明

输入内容包括在命令行上放置的或从标准输入读取的一个或多个对象或归档库 *files* （请参阅 *ar*(1) ）。标准输出是对象文件名称对的列表，意味着对中的第一个文件引用在第二个文件中定义的外部标识符。输出可以由 **tsort** 处理，以查找适合于 **ld** 一遍访问的库排序（请参阅 *tsort*(1) 和 *ld*(1) ）。请注意，链接编辑器 **ld** 能够多遍访问归档格式的归档且在构建归档时不要求使用 **lorder** 。但是，在链接编辑过程中使用 **lorder** 命令可能允许稍微更有效地访问归档。

ar 维护的符号表允许 **ld** 随机访问归档中的符号和文件，这样在构建归档库时就不需要使用 **lorder** （请参阅 *ar*(1)）。

外部语言环境影响

环境变量

下列国际化变量会影响 **lorder** 的执行：

LANG 用于确定没有 **LC_ALL** 和其他 **LC_*** 环境变量时本地语言、本地惯例和编码字符集的语言环境类别。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值 **C** （请参阅 *lang*(5) ），而不是使用 **LANG** 。

LC_ALL
用于确定所有语言环境类别的值，它优先于 **LANG** 和其他 **LC_*** 环境变量。

LC_COLLATE
用于确定字符排序的语言环境类别。

LC_CTYPE
用于确定字符处理函数的语言环境类别。

LC_MESSAGES
用于确定应该用来影响写入标准错误的诊断消息的格式和内容的语言环境。

LC_NUMERIC
用于确定数字格式的语言环境类别。

NLSPATH
用于确定消息清单的位置，以便处理 **LC_MESSAGES** 。

如果任一国际化变量包含无效设置，则 **lorder** 就会认为所有国际化变量都设置为 **C** 。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

从现有 **.o** 文件构建新库：

ar cr library ‘lorder *.o | tsort‘

如果创建的库具有如此多的对象，以致于 Shell 无法正确处理 ***.o** 扩展，则以下方法可能是很有用的：

ls | grep ’.o$’ | lorder | tsort | xargs ar cq library

警告

其名称不以 **.o** 结尾的对象文件将被忽略，即使包含在库归档中。其全局符号和引用归入某个其他文件。

文件

/var/tmp/*symref 临时文件
/var/tmp/*symdef

另请参阅

系统工具：

ar(1) 创建归档库
ld(1) 调用链接编辑器

其他：

tsort(1) 生成项目的排序列表（拓扑排序）

符合的标准

lorder：SVID2、SVID3、XPG2、XPG4

名称

lp、lpalt、cancel - 打印/更改/取消 LP 目标上的请求

概要

lp [**-c**] [**-d***dest*] [**-m**] [**-n***number*] [**-o***option*] [**-p***priority*] [**-s**] [**-t***title*] [**-w**] [*file* ...]

lpalt *id* [**-d***dest*] [**-i**] [**-m**] [**-n***number*] [**-o***option*] [**-p***priority*] [**-s**] [**-t***title*] [**-w**]

cancel [*id* ...] [*dest* ...] [**-a**] [**-e**] [**-i**] [**-u***user*] [**-f**]

说明

lp 命令将要打印的文件队列。 **lpalt** 命令更改队列请求中的信息。 **cancel** 命令删除队列的请求。

lp 命令

lp 命令安排命名文件 *file* ... 及相关信息（统称 *request* ）队列，以输出到 LP（行式打印机）子系统中的目标。无论实际的输出设备是什么，该过程均称作打印。

lp 将唯一的标识符与每个请求相关联，并使用以下消息将其写入标准输出：

request id is *dest***-***sequence* **(***fileinfo***)**

请求 ID 是 *dest***-***sequence*，随后可以使用它来更改、取消请求或查找请求的状态（请参阅下文的 **lpalt** 和 **cancel** 以及 *lpstat*(1) ）。

例如，在以下消息中，

request id is pr47lf8e-2410 (1 file)

请求 ID 是 **pr47lf8e-2410** 。

lp 选项和参数

lp 可识别下列选项和参数。关键字选项可以按任意顺序指定。 *file* ... 名称必须位于最后。

file ... 打印各命名文件。如果没有指定文件名，则假定为标准输入。连字符符号 **(-)** 也指定标准输入，可以在命令行上与文件名混合使用。如果指定多个 **-** ，则忽略除第一个之外的所有连字符。文件将按照其指定顺序打印。最多可以指定 832 个文件名。

-c 将命名的文件复制到 LP 子系统假脱机目录。

通常，文件将链接到一个假脱机目录。所链接文件的所有权和模式将保持不变。如果给定 **-c** 选项或者无法进行链接（可能因为文件与假脱机目录不驻留在相同的文件系统上），文件将复制到假脱机目录中。副本的所有权和模式设置为允许所有者 **lp** 的读取访问权限，而只允许组 **lp** 的读取权限。

如果链接而不是复制文件，在发出请求之前但在其打印之后对命名文件做出的任何更改将在打印的输出中得到反映。始终会复制而不是链接标准输入。

-d*dest* 选择 *dest* 作为将执行打印的打印机或打印机类。如果 *dest* 是打印机，则仅在该特定的打印机上打印该请求。如果 *dest* 是类，则请求将在作为该类成员的第一个可用打印机上打印。某些情况

下（打印机不可用，文件空间有限等），可能无法接受对特定 *dest* 的请求（请参阅 *accept*(1M) 和 *lpadmin*(1M)）。

如果省略 **-d** 选项，则将从环境变量 **LPDEST** 中采用 *dest* 。如果该变量未设置或者为空，则将从环境变量 **PRINTER** 中采用 *dest* 。如果该变量未设置或者为空，则使用缺省的队列。如果不存在缺省队列，或者缺省队列虽存在，但却为空或包含无效的目标条目（或者 **LPDEST** 已设置但却无效， **PRINTER** 已设置但却无效）， **lp** 将发出错误消息，而请求不会排队。打印机和类名以及缺省队列由 LP 子系统管理员来定义（请参阅 *lpadmin*(1M) 和 *lpstat*(1)）。

-m 在打印请求后向用户发送邮件消息（请参阅 *mail*(1)）。缺省情况下，正常完成打印请求时将不发送邮件。

-n*number* 打印输出的副本 *number* 。如果用该选项指定了非法的副本 *number* ，则缺省的副本数目为 1。

-o*option* 指定打印机相关 *option* 。可以通过重复 **-o** 选项来指定多个打印机选项。有关系统支持的打印机的可用选项的信息，请参阅 **/etc/lp/interface** 目录内接口脚本的打印机名称。

-p*priority* 设置打印请求的优先级。 *priority* 必须介于 0（优先级最低）到 7（优先级最高）的范围内。优先级由 **lpsched** 调度程序用来为目标打印机或打印机类选择下一个假脱机文件。如果优先级小于 *fence* （打印机的最低优先级），则打印请求将延迟到降低阻隔或提高优先级。打印机队列的缺省值是由 **lpadmin** 或 **lpfence** 命令设置的缺省优先级（请参阅 *lpadmin*(1M) 和 *lpsched*(1M)）。类队列的缺省值是类中打印机的最高缺省优先级。

-s 禁止标准输出消息（如“**request id is** ...”）通过 **lp** 打印。错误消息仍在标准错误上显示。

-t*title* 在输出的横幅页打印 *title* 。 *title* 的最大长度为 79 个字节。长度大于 79 个字节的 *title* 将被截断至 79 个字节。

-w 在打印请求后向用户的终端写入消息。如果用户未登录或用户拒绝向其终端发送消息（请参阅 *mesg*(1)）或者（对于远程打印）如果 **rlpdaemon** （请参阅 *rlpdaemon*(1M)）未在用户的本地系统上运行，则将发送邮件。

lpalt 命令

lpalt 命令更改先前 **lp** 命令发出的请求（如果当前未打印）。（要将当前正在打印的请求重新排队，请使用 **disable** 命令（请参阅 *enable*(1)）停止打印机）。

lpalt 选项

lpalt 可识别以下选项和参数，它们可以按任意顺序指定。关键字及其参数之间不能使用空白字符。

ID 指定要更改的请求。 *ID* 是 **lp** 或 **lpalt** 返回的请求 ID 。

-d*dest* 将对命名打印机或类 *dest* 的请求重新排队。新的唯一请求 ID 将写入标准输出。

-i 仅更改本地请求。

-m 在正常完成打印请求时发送邮件。

-n*number* 将副本数目更改为 *number* 。

-o*option* 指定打印机相关 *option* 。可以通过重复 **-o** 选项来指定多个打印机选项。将删除该请求 ID 的先前 **lp** 和 **lpalt** 命令的所有 **-o** 选项。

-p*priority* 将请求的优先级更改为 *priority* 。

-s 禁止标准输出消息（如“**new request id is** ...”）通过 **lpalt** 打印。错误消息仍在标准错误上显示。

-t*title* 更改输出横幅页上的 *title* 。

-w 在打印请求后向用户的终端写入消息。如果用户未登录或用户拒绝向其终端发送消息（请参阅 *mesg*(1)）或者（对于远程打印）如果 **rlpdaemon** （请参阅 *rlpdaemon*(1M)）未在用户的本地系统上运行，则将发送邮件。

cancel 命令

cancel 命令取消用 **lp** 命令发出的请求（即使它们正在打印）。

通过取消当前打印的请求，可以释放打印机，以打印下一个可用的请求。

cancel 选项和参数

cancel 可识别以下选项和参数，它们可以按任意顺序指定。关键字及其参数之间不能使用空白字符。当 cancel 与不同选项和参数的组合一起使用时，它将首先处理 *ID* ...，接着处理 *dest* ...，再接着处理 **-a** ，然后处理 **-e** ，最后处理 **-u** ，而不管在命令行中指定选项和参数的顺序。

ID ... 指定一个或多个要取消的请求。 *ID* 是 **lp** 或 **lpalt** 返回的请求 ID 。

dest ... 指定一个或多个打印机或打印机类。如果未指定 **-a** 、 **-e** 或 **-u** 选项，则取消每个 *dest* 上正在打印的请求。这种情况下， *dest* 必须是打印机，而不能是类。如果指定了 **-a** 、 **-e** 或 **-u** 选项，则指定在其上执行相应取消操作的目标。这种情况下， *dest* 可以是打印机或类。

-a 删除用户在每个 *dest* 上拥有的所有请求，如果未指定 *dest* 而指定了 **-f** 选项，则删除用户在系统中所有目标上拥有的所有请求。请求的所有者取决于用户的登录名和用来调用 **lp** 命令的计算机的主机名。

-e 清空每个 *dest* 的假脱机队列，如果未指定 *dest* 而指定了 **-f** 选项，则清空系统中所有目标的假脱机队列。只有拥有适当特权的用户才能使用该选项。

-i 仅取消本地请求。

-u*user* 删除每个 *dest* 上属于 *user* 的所有请求，如果未指定 *dest* 而指定了 **-f** 选项，则删除系统中所有目标上属于 *user* 的所有请求。可以通过重复 **-u** 选项来指定更多用户。只有拥有适当特权的用户才能使用该选项。

-f 强制取消 **-a/-e/-u** ，以便对系统中的所有目标执行操作。

打印概述

打印机可以打印一个或两个目标队列中的请求：它自己的专用队列和可选的类队列，后者可以使用一台或多台打印机。目标队列使用 **lpadmin** 命令设置。 **lp** 命令按照用户的指示将打印请求放入打印机或类目标队列。 **lpsched** 调度程序将目标队列中的请求定向到打印机。 **accept** 和 **reject** 命令控制 **lp** 是否可以将请求放入目标队列。 **enable** 和 **disable** 命令控制 **lpsched** 是否可以将排队的请求发送到打印机。如果打印机有两个队列，并且一个队列拒绝请求，用户仍可以将请求定向到另一个目标队列来打印这些请求。 **lpstat** 报告目标队列和调度程序的当前状态。请参阅 *enable*(1) 、 *lpstat*(1) 、 *accept*(1M) 和 *lpadmin*(1M) 。

外部语言环境影响

环境变量

LANG 确定当 **LC_ALL** 和相应的环境变量（以 **LC_** 开头）都未指定语言环境时，语言环境类别将使用的语言环境。如果未设置 **LANG** 或将其设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5) ）。

LC_ALL 确定用于覆盖各语言环境类别的任何值（通过设置 **LANG** 或以 **LC_** 开头的任何环境变量来指定）的语言环境。

LC_CTYPE 用于确定将语言环境文本字节序列解释为何种字符（例如参数和输入文件中单字节字符和多字节字符）。

LC_MESSAGES 用于确定显示消息的语言。

LPDEST 确定输出设备或目标。如果未设置 **LPDEST** 环境变量，则使用 **PRINTER** 环境变量。 **-d** *dest* 选项优先于 **LPDEST** 。

PRINTER 确定输出设备或目标。如果未设置 **LPDEST** 和 **PRINTER** 环境变量，则使用缺省队列。 **-d** *dest* 选项和 **LPDEST** 环境变量优先于 **PRINTER** 。

如果任一国际化变量包含无效设置，命令将认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

退出值包括：

0	成功完成。
>0	发生了错误。

举例

对于用接口脚本（它定义了 **-c** 选项，使得打印机在压缩模式下打印）配置的激光打印机 **lp2** ，使用以下命令在 **lp2** 上以压缩打印模式打印 **myfile** ：

```
lp -dlp2 -oc myfile
```

lp 可以在管道线的末端用来打印先前命令的结果。它通常与 **pr** 命令（请参阅 *pr*(1) ）一起使用来打印格式化的输出。对于缺省的打印机，要将文件 **.profile** 格式化成页，并打印三个副本：

```
pr .profile | lp -n3
```

警告

只有从发出初始 **lp** 命令的系统上并且在为指定的打印机启用限制取消功能（请参阅 *lpadmin*(1M)）时，才能取消远程打印请求；属于该打印机的请求只能由管理员或请求它的用户来取消。

远程打印请求只能从发出初始 **lp** 命令的系统上，由管理员或请求它的用户来取消。如果该更改请求正在打印，远程系统将忽略该请求。

对于远程系统， **lpalt** 无法更改 *dest* 和 *priority* 。

有关目标队列和打印请求的信息在 **/var/spool/lp** 目录下的 **pstatus** 、 **qstatus** 和 **outputq** 文件中进行维护。由于这些文件中存储的数据的格式可能在将来更改，因此这些文件不应该由 LP 子系统之外的其他任何应用程序直接读取。

文件

/etc/lp	假脱机配置数据的目录
/etc/lp/interface	活动 LP 设备接口脚本的目录
/usr/lib/lp	模型和字体文件目录的目录
/var/adm/lp	假脱机日志文件的目录
/var/spool/lp	LP 假脱机文件和目录的目录

另请参阅

enable(1)、 lpstat(1)、 mail(1)、 slp(1)、 accept(1M)、 lpadmin(1M)、 lpana(1M)、 lpsched(1M)、 rcancel(1M)、 rlp(1M)、 rlpdaemon(1M)、 rlpstat(1M)。

符合的标准

lp：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

cancel：SVID2、SVID3、XPG4

名称

lpfilter、divpage、fontdl、lprpp、plotdvr、printstat、reverse - lp 接口脚本调用的过滤器

概要

/usr/lbin/divpage [**-p** | **-l**] [**-h** | **-q**] [**-n***FontID*] *filename*

/usr/lbin/fontdl [**-n***FontID*] [**-l**] [**-p**] *filename*

/usr/lbin/lprpp [**-i**] [**-o**] [**-e**] [**-l***nn*] [**-n**] [**-p**]

/usr/lbin/plotdvr **-l** *request_id* **-u** *username* [**-e**] [**-f**] [**-i**] *filename*

/usr/sbin/printstat **-l** *request_id* **-u** *username* *filename*

/usr/sbin/reverse [**-l** *page_length*]

备注

这些过滤器的结构目前正在审核之中。在将来的发行版中，它们可能会过时或重新构建。

说明

lp 子系统使用各种过滤器来获得特定类型的设备或数据的专门行为。此条目描述了当前所支持的过滤器。

许多过滤器使用特定 *username* 和 *filename* 来确定生成打印消息的用户的位置。

filename 用于确定生成请求的系统的 *hostname* ，其形式必须为 [*dirname*]**/d?A** *nnnhostname* 或 [*dirname*]**/dA** *nnnnhostname* ，其中 *dirname* 不是路径名，而只是基名的父目录名称。当 *filename* 在打印机接口脚本中被设为 **$6** 时，即可达到该要求。

divpage

具备每版打印多页功能及内置字体选择功能。

选项:

-p 将模式设置为纵向（缺省情况下）。

-l 将模式设置为横向。

-h 打印半页（缺省情况下）。

-q 打印四分之一页。

-n *FontID* 使用字体号 *FontID* 。缺省值为 0。使字符串 **^{E}C(** *FontID***X** 被发送到打印机。

fontdl

fontdl 将 *filename* 中包含的字体下载到与标准输出相连的打印机。

选项:

-n *FontID* 指定要引用的字体的 ID 号。缺省值为 0。

-l 指定横向模式。缺省情况下为纵向。

-p 指定比例间隔。缺省情况下是固定的。

lprpp

一种过滤器，用于将退格叠印转换为行套印，以水平打印定位来增强粗体打印。像 LaserJet 之类的打印机需要此功能，此类打印机不能通过叠印来生成粗体打印。

选项：

-i 将 <ANYCHAR> 转换为 <ANYCHAR><BACKSPACE>_，以使 ANYCHAR 变成斜体。也可以将叠印（粗体）字符准确地变成斜体。在“散列叠印”的情况下 无效，例如：

<ANYCHAR><BACKSPACE><DIFFERENTCHAR><BACKSPACE>_

-o 只打印奇数页。与 **-e** 一起使用，用于双面打印。

-e 仅打印偶数页。与 **-o** 一起使用，用于双面打印。

-l *nn* 指定页长度，以行数来计。如果没有选择 *-n* 或 *-p*，则缺省值为 60；否则，缺省值为 66。

-n 为 *nroff* 命令的打印输出指定 nroff 模式。每页打印 66 行，第一行出现在打印机的逻辑行 4。

-p 在 *pr* 命令中，为打印输出指定 pr 模式。每页打印 66 行，第一行出现在打印机的逻辑行 3。

plotdvr

HP-GL 绘图仪过滤器。此过滤器对数据进行扫描，以查找“PG”命令（送纸）。当发现此数据时，过滤器将其从数据流中去除，并且通知正在请求的用户，他们需要在绘图仪中更换纸张。

选项：

-l *request_id* 指定打印机请求 ID，并在各种关于绘图请求的消息中使用。

-u *username* 请求用户的登录名，用于与请求相关的用户进行通信。

-e 指定转义序列的用法，而不是指定 HP-GL 命令的用法，以确定绘图仪状态。

-f 绘图时无须停下来更换纸张。未从数据流中去除“PG”命令，并且用户也未得到相应的通知。此选项用于能够自动换页的绘图仪。

-i 防止绘图仪初始化。

printstat

询问 RS232 打印机的状态，并且直到打印机就绪时才返回。如果打印机为 *off-line*（脱机）、*out of paper*（缺纸）或 *disconnected*（断开连接），则将这种情况定期提交给打印请求的提交者，直到这种情况得以解决。当打印机准备打印时，退出该命令。

必须将标准输入和标准输出全部连接到串行打印机设备。

此程序使用 发送状态命令 **^{E}C?^{D}1?** 确定状态。并非所有的串行打印机均响应此命令。只有下列配置支持此命令：

打印机	注释
LaserJet	不支持
LaserJetII	支持
LaserJetIID	要求 HP 26013A 模块
LaserJetIIP	不支持
LaserJetIII	要求 HP 26013A 模块
LaserJet2000	不支持

选项：

-l *request-id* 在与用户进行的各种通信中使用的打印请求 ID。

-u *username* 请求用户的登录名，用于与请求相关的用户进行通信。

reverse

按相反的页码顺序将出现在标准输入中的数据打印到标准输出中。此命令最多可以处理 2000 页。

Options:

-l *page_length* 指定页长度，以行数来计。缺省值为 66。

诊断信息

错误消息和诊断消息可显示在打印输出上，可显示在用户终端上，也可邮寄给用户，这要视具体情况而定。

警告

这些过滤器的接口几乎都不一致。

另请参阅

lp(1)、lpadmin(1M)。

《管理系统和工作组》。

名称

lpstat - 报告 LP 子系统的状态信息

概要

lpstat [**-drst**] [**-a**[*list*]] [**-c**[*list*]] [**-o**[*list*]] [**-p**[*list*]] [**-u**[*list*]] [**-v**[*list*]] [**ID**]...

说明

lpstat 实用程序将有关 LP 子系统当前状态的信息写入到标准输出。

如果没有指定参数，则 **lpstat** 写入仍在输出队列中的用户对 **lp** 的所有请求的状态。

选项

lpstat 实用程序支持 XBD 规范（《Utility Syntax Guidelines》的第 10.2 节），只是选项参数是可选的，并且不能作为独立的参数提供。

下面一些选项的后面可以紧跟一个可选列表，可以是以下两种形式之一：用逗号分隔的项列表，或者用逗号或一个或多个空白字符分隔的带引号的项列表。请参阅“举例”。

如果在以下选项后省略列表，则会使与该选项有关的所有信息写入到标准输出中；例如：

lpstat -o

写入仍在输出队列中的所有输出请求的状态。

-a[*list*] 为输出请求写入目的地的接受状态。 *list* 参数是混合打印机名称和类名称的列表。

-c[*list*] 写入类名称及其成员。 *list* 参数是类名称的列表。

-d 为输出请求写入系统缺省的目的地。 **lpstat** 检查环境变量 **LPDEST** 中缺省的目的地。如果该变量未设置或为空，则 **lpstat** 检查环境变量 **PRINTER** 中缺省的目的地。如果该变量未设置或为空，则 **lpstat** 检查缺省队列中缺省的目的地。

-o[*list*] 写入输出请求的状态。 *list* 参数是混合打印机名称、类名称和请求 ID 的列表。

-p[*list*] 写入打印机的状态。 *list* 参数是打印机名称的列表。

-r 写入 LP 请求调度程序的状态。

-s 写入状态摘要，包括 LP 调度程序的状态、系统缺省的目的地、类名称及其成员的列表以及打印机及其相关设备的列表。

-t 写入所有状态信息。

-u[*list*] 写入用户输出请求的状态。 *list* 参数是登录名的列表。 *list* 中允许的最大登录名数为 50。

-v[*list*] 写入打印机名称及其相关设备的路径名。列表参数是打印机名称的列表。

操作数

支持下列操作数：

ID 写入 **ID** 的输出请求的状态。 **ID** 是请求 ID，由 **lp** 返回。

dest 写入 **dest** 的输出请求的状态。 **dest** 是打印机名称或类名称。

STDIN

未使用。

输入文件

无。

环境变量

下列环境变量影响 **lpstat** 的执行：

LANG 为没有设置或设置为 Null（空）的国际化变量提供缺省值。如果 **LANG** 未设置或设置为 Null（空），则将使用特定于实现的缺省语言环境中的相应值。如果任何国际化变量包含无效设置，则实用程序认为尚未定义任何变量。

LC_ALL 如果设为非空字符串值，则会覆盖所有其他国际化变量的值。

LC_CTYPE 确定将语言环境文本数据字节序列解释为何种字符（例如参数中的单字节和多字节字符）。

LC_MESSAGES 确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息以及写入到标准输出的信息消息的格式和内容。

LC_TIME 在使用 **-a** 、 **-o** 、 **-p** 、 **-t** 或 **-u** 选项显示行式打印机状态信息时，确定日期和时间字符串输出的格式。

NLSPATH 确定消息目录的位置，以便处理 **LC_MESSAGES** 。

TZ 确定日期和时间字符串使用的时区。

LPDEST 确定输出设备或目的地。如果未设置 **LPDEST** 环境变量，则使用 **PRINTER** 环境变量。

PRINTER 确定输出设备或目的地。如果未设置 **PRINTER** 环境变量，则使用缺省队列。 **LPDEST** 环境变量优先于 **PRINTER** 。

异步事件

缺省值。

STDOUT

标准输出是一个文本文件，其中包含“选项”中说明的信息，这些信息未指定格式。

STDERR

仅用于诊断消息。

输出文件

无。

扩展说明

无。

退出状态

返回下列退出值:

0 成功完成。

>0 发生错误。

错误结果

缺省值。

实际应用信息

lpstat 实用程序无法可靠地确定所有可能的环境中打印请求的状态。当打印机在另一个操作系统的控制之下，或者驻留在跨网络的远程系统中时，它可能不需要在离开本地操作系统控制之后确定打印作业的状态。即使在本地打印机上，后台处理打印机中的硬件时，也可能会出现在打印最后一页之前早就完成了打印作业的情况。

举例

1. 获取两台打印机的状态、两台打印机的路径名、所有类名称的列表以及名为 HiPri-33 的请求的状态:

   lpstat -plaser1,laser4 -v"laser2 laser3" -cHiPri-33

2. 使用过时的混合空白和逗号形式来获取用户打印作业状态:

   lpstat -u"ddg,gmv, maw"

未来方向

在将来的版本中可能会引入完全支持 XBD 规范（《Utility Syntax Guidelines》的第 10.2 节）的 **lpstat** 版本。

另请参阅

cancel(1)、lp(1)。

更改历史记录

在第 2 期中首次发布。

第 3 期

注释了该实用程序以 8 位透明方式的操作。

说明了该实用程序在国际化环境中的操作。

第 4 期

重新组织的格式。

对符合《Utility Syntax Guidelines》的例外情况进行了说明。

必须支持国际化环境变量。

符合的标准

lpstat：SVID2、SVID3、XPG2、XPG3、XPG4

HP-UX 扩展

说明

任何不是 *options* 的参数都被假定为请求 *id* （由 **lp** 返回）或 LP 目的地。 **lpstat** 打印与此类请求 *id* 对应的请求的状态，或者输出属于这些目的地的请求的状态。 *options* 可以以任何顺序出现，可以重复，并可与其他参数混用。

-i 禁止报告远程状态。

-o[*list*] 另请参阅 **-i** 选项。

-t 输出所有状态信息。与指定 **-r** 、 **-s** 、 **-a** 、 **-p** 、 **-o** 相同，请参阅 **-i** 选项。

安全限制

只有具有 **lp** 子系统授权或 **printqueue** 二级子系统授权的用户才可以查看整个队列。未经授权的用户只可以查看其自己的作业，其敏感度级别由用户的当前敏感度级别控制。

allowmacaccess 权限允许查看更高敏感度级别的作业。

举例

检查作业是否在队列中：

lpstat

检查队列中作业的相对位置：

lpstat -t

验证作业调度程序是否正在运行：

lpstat -r

文件

/var/spool/lp/*
/var/adm/lp/*
/etc/lp/*
/usr/lib/lp/*

另请参阅

enable(1)、 lp(1)、 rlpstat(1M)。

符合的标准

lpstat： SVID2、SVID3、XPG2、XPG3、XPG4

名称

ls、lc、l、ll、lsf、lsr、lsx - 列出目录的内容

概要

ls [**-abcdefgilmnopqrstuxACFLR1**] [*names*]

lc [**-abcdefgilmnopqrstuxACFLR1**] [*names*]

l [*ls_options*] [*names*]
ll [*ls_options*] [*names*]
lsf [*ls_options*] [*names*]
lsr [*ls_options*] [*names*]
lsx [*ls_options*] [*names*]

说明

对于每个目录参数， **ls** 命令列出目录的内容。对于每个文件参数， **ls** 重复其名称和请求的任何其他信息。缺省情况下，输出按升序排序（请参阅下面的“环境变量”）。如果未指定参数，则列出当前目录。如果指定了几个参数，则首先对参数进行适当排序，但是文件参数出现在目录及其内容之前。

如果您是具有相应特权的用户，则缺省情况下将列出除 **.** 和 **..** 之外的所有文件。

有三种主要列出格式。所选格式取决于是否要输出到登录设备（由输出设备文件是否为 **tty** 设备确定），也可能受选项标志的控制。登录设备的缺省格式是以多列格式列出目录的内容，其中的条目按列垂直排序。（当单独的文件名（与目录名相对）出现在参数列表中时，那些文件名始终在列中沿页的横向而不是纵向进行排序，因为单独的文件名可以是任意长度的）。如果标准输出不是登录设备，则缺省格式为每行列出一个条目。 **-C** 和 **-x** 选项启用多列格式， **-m** 选项启用流式输出格式（其中文件沿页的横向列出，用逗号分隔）。为了确定 **-C** 、 **-x** 和 **-m** 选项的输出格式， **ls** 使用环境变量 **COLUMNS** 确定在每个输出行上可用的字符位置数。如果未设置此变量，则使用 **terminfo** 数据库并根据环境变量 **TERM** 来确定列数。如果无法获得此信息，则假定为 80 列。

命令 **lc** 的功能与 **ls** 相同，只是 **lc** 的缺省输出是分列的，即使已重定向输出。

选项

ls 可识别下列选项:

-a 列出所有条目；通常不列出其名称以句点 (**.**) 开头的条目。

-b 以八进制 *****ddd* 表示法列出非输出字符。

-c 将 i 节点的上次修改时间（创建文件、更改模式等）用于排序 (**-t**) 或输出 (**-l** （字母 L）。

-d 如果参数为目录，则仅列出其名称（而不是其内容）；通常与 **-l** （字母 L）一起使用以获取目录的状态。

-e 列出文件的盘区属性。如果任何文件具有盘区属性，则此选项列出盘区大小、保留的空间和分配标志。此选项必须与 **-l** （字母 L）选项一起使用。

-f 将每个参数都解释为目录，并列出在每个插槽中找到的名称。此选项禁用 **-l** （字母 L）、 **-r** 、 **-s** 和 **-t** ，但启用 **-a** ；顺序为条目在目录中出现的顺序。

-g 与 **-l** （字母 L）相同，只是仅输出组（省略所有者）。如果同时指定了 **-l** （字母 L）和 **-g** ，则不输出所有者。

-i 对于每个文件，在报告的第一列中列出 i 节点编号。在多列输出中使用时，该编号在每列中的文件名之前。

-l （字母 L）按长格式列出，提供每个文件的模式、链接数、所有者、组、大小（字节）和上次修改时间（另请参阅下面的“说明”和“访问控制列表”）。如果上次修改时间离现在超过六个月或者是将来的任何时间，则用年份替换修改时间的小时和分钟。如果文件是专用文件，则大小字段包含主次设备编号而不是大小。如果文件为符号链接，则输出后跟 **->** 以及引用文件的路径名的文件名。

-m 流式输出格式。

-n 与 **-l** （字母 L）相同，只是输出所有者的 UID 和组的 GID 编号，而不是关联的字符串。

-o 与 **-l** （字母 L）相同，只是仅输出所有者（省略组）。（如果同时指定了 **-l** （字母 L）和 **-o**，则不输出组）。

-p 如果文件是目录，则在每个文件名之后放置斜线 (**/**) 。

-q 将文件名中的非输出字符作为字符 (**?**) 列出。

-r 根据需要，反转排序顺序以获得反向（降序）排序或首先获得最旧的条目。

-s 列出每个条目的大小（以块为单位），其中包括间接块。列出的第一个条目是目录中的总块数。在多列输出中使用时，块数在每列中的文件名之前。文件中的间接块数与文件系统相关。

-t 在按字母顺序排序之前，按修改时间进行排序（最晚的在最前）。

-u 将上次访问时间而不是上次修改时间用于排序（ -t 选项）或输出（ -l （字母 L）选项）。

-x 列出多列输出，沿页的横向而不是纵向对条目排序。

-A 与 **-a** 相同，只是不列出当前目录 **.** 和父目录 **..** 。对于具有相应特权的用户，此标志缺省为打开，它由 **-A** 关闭。

-C 列出多列输出，沿列的纵向对条目排序。

-F 在每个文件名之后，放置以下项之一:

+ 如果文件为目录或指向目录的符号链接，则放置斜线 (**/**) 。
- 如果文件是可执行的，则放置星号 (*****) ；
- 如果文件是指向某个文件的符号链接，则放置 at 符号 (**@**) ；
- 如果文件是 fifo，则放置竖线 (**|**) 。

-L 将所有符号链接的文件信息和文件类型（不管是在命令行上指定还是在文件层次结构中遇到）评估为链接所引用的文件的而不是链接本身的。但是， **ls** 将写入链接本身的名称，而不是链接所引用的文件的名称。将 **-L** 与 **-l** 一起使用时，将以长格式写入符号链接的内容。

-R 以递归方式列出遇到的子目录。

-1 （一）以单列格式列出文件名，而不管使用什么输出设备。这将强制用户终端使用单列格式。

在下列相互排斥对中指定其中的多个选项不视为错误： **-C** 和 **-l** （字母 L）、 **-m** 和 **-l** （字母 L）、 **-x** 和 **-l** （字母 L）、 **-C** 和 **-1** （一）以及 **-c** 和 **-u** 。

对于各种不同格式，几个简写版本名称可识别 **ls** ：

l 等效于 **ls -m**
ll 等效于 **ls -l** （字母 L）
lsf 等效于 **ls -F**
lsr 等效于 **ls -R**
lsx 等效于 **ls -x**

简写符号是作为指向 **ls** 的链接实现的。简写版本的选项参数的行为与将上面的长格式和其他参数一起使用时完全相同

模式位解释（**-l** 选项）

在 **-l** （字母 L）选项生成的列表中输出的模式包括 10 个字符，例如， **-rwxr-xr-x** 。

第一个字符指示条目类型：

b 块设备专用文件
c 字符设备专用文件
d 目录
l 符号链接
n 网络设备专用文件
p Fifo（也称为“命名管道”）设备专用文件
s 插槽
- 普通文件

接下来的 9 个字符是按每组包括三个字符的三组字符解释的，每组字符标识所有者、组和其他人类别的访问和执行权限，如 *chmod*(1) 中所述。 **-** 指示未授予权限。可以按任意组合将各种权限放置在一起，例外情况是： **x**、 **s**、 **S**、 **t** 和 **T** 字符是相互排斥的，如下所示。

-r-------- 由所有者读取
--w------- 由所有者写入
---x------ 由所有者执行（或搜索目录）；在执行时不设置用户 ID
---s------ 由所有者执行（或搜索）；在执行时设置用户 ID

---S------	禁止所有者执行（或搜索）；在执行时设置用户 ID
----r-----	由组读取
-----w----	由组写入
------x---	由组执行（或搜索）；在执行时不设置组 ID
------s---	由组执行（或搜索）；在执行时设置组 ID
------S---	禁止组执行（或搜索）；在执行时设置组 ID
-------r--	由其他人读取
--------w-	由其他人写入
---------x	由其他人执行（或搜索）；在执行时不设置粘着位
---------t	由其他人执行（或搜索）；在执行时设置粘着位
---------T	禁止其他人执行（或搜索）；在执行时设置粘着位

对模式字符的解释如下所示:

- 拒绝对应位置中的所有权限。

r 将读取权限授予对应的用户类。

w 将写入权限授予对应的用户类。

x 将执行（或在目录中搜索）权限授予对应的用户类。

s 将执行（搜索）权限授予对应的用户类。像所有者（设置用户 ID，SUID）或组（设置组 ID，SGID）那样执行文件，如位置所示。

S 拒绝将执行（搜索）权限授予对应的用户类。像所有者（设置用户 ID，SUID）或组（设置组 ID，SGID）那样执行文件，如位置所示。

t 将执行（搜索）权限授予其他人。将设置“粘着”（保存文本图像）位（请参阅 *chmod*(2) 中对 **S_ISVTX** 的说明）。

T 拒绝将执行（搜索目录）权限授予其他人。将设置“粘着”（保存文本图像）位。

如果指定的选项（如 **-s** 或 **-l** （字母 L）选项）导致列出以字节或块为单位目录和（或）文件大小，则也会在列表的开头输出总块数（包括间接块）。

访问控制列表 (ACL)

如果文件具有可选的 ACL 条目，则 **-l** （字母 L）选项在文件权限的后面显示一个加号 **(+)** 。显示的权限是文件访问控制列表的摘要表示，如 **st_mode** 字段中的 **stat()** 返回的那样（请参阅 *stat*(2) ）。要列出访问控制列表的内容，请使用 **lsacl** 命令（请参阅 *lsacl*(1) 和 *acl*(5) ）或 **getacl** 命令（请参阅 *getacl*(1) 和 *aclv*(5) ），这两个命令分别适用于 HFS 文件系统和 JFS 文件系统。

外部语言环境影响

环境变量

如果设置了 **COLUMNS** 变量，则 **ls** 将使用在确定分列输出位置时提供的宽度。

LANG 确定当 **LC_ALL** 和相应的环境变量（以 **LC_** 开头）都未指定语言环境时，语言环境类别将使用的语言环

境。如果未设置 **LANG** ，或者其值为空，则会缺省为 **C** （请参阅 *lang*(5) ）。

LC_COLLATE 用于确定输出的排序顺序。

LC_CTYPE 用于确定将哪些字符归为 **-b** 和 **-q** 选项的非输出字符，以及如何解释文件名内的单字节和（或）多字节字符。

LC_TIME 用于确定由 **-g** 、 **-l** （字母 L）、 **-n** 和 **-o** 选项输出的日期和时间字符串。

LC_MESSAGES 用于确定显示消息（日期和时间字符串除外）的语言。

如果任一国际化变量包含无效设置，则所有国际化变量都缺省为 **C** （请参阅 *environ*(5) ）。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

ls 退出时返回下列值之一：

0 已成功列出所有输入文件。

>0 **ls** 已异常中止，因为访问文件时出现错误。下列情况导致了错误：

- \+ 找不到指定的文件。
- 用户没有读取目录的权限。
- 进程无法获取足够的内存。
- 指定的选项无效。

举例

输出当前工作目录中所有文件的长列表（包括文件大小）。首先列出最近修改（最晚）的文件，后跟修改时间稍早的文件，依此类推，最后列出最早修改的文件。还输出其名称以 **.** 开头的文件。

ls -alst

警告

根据输出是否为登录 (*tty*) 设备来设置选项是不合需要的，因为 **ls -s** 与 **ls -s | lp** 有很大差异。另一方面，如果不使用此设置，则会导致使用 **ls** 的旧 Shell 脚本几乎总是失败。

文件名中的非输出字符（无 **-b** 或 **-q** 选项）可能会导致分列输出对不齐。

相关内容

NFS

-l （字母 L）选项在联网文件的访问权限位的后面不显示表示存在可选访问控制列表条目的加号 **(+)** 。

作者

ls 由 AT&T、加州大学伯克利分校和 HP 联合开发。

文件

/etc/group	用于 **-l** （字母 L）和 **-g** 的组 ID。
/etc/passwd	用于 **-l** （字母 L）和 **-o** 的用户 ID。
/usr/share/lib/terminfo/?/*	用于终端信息。

另请参阅

chmod(1)、 find(1)、 getacl(1)、 lsacl(1)、 stat(2)、 acl(5)、 aclv(5)。

符合的标准

ls：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

lsacl - 列出文件的访问控制列表 (ACL)

概要

/usr/bin/lsacl [**-l**] *file*...

说明

lsacl 以符号“短”格式列出一个或多个文件的访问控制列表 (ACL)，在输出结果中一个文件的 ACL 占用一行，后接文件名；请参阅 *acl*(5) 以了解 ACL 语法。

选项

lsacl 可识别下列选项：

-l 以长格式输出 ACL。每个文件的 ACL 长度可以超过一行，并且开头始终是文件名、冒号和新行。连续的文件名之间用空行分隔。

如果将文件名参数指定为连字符 (**-**) ， **lsacl** 会输出标准输入的 ACL 。缺省情况下，它将文件名输出为 **-** 。对于 **-l** 选项则输出文件名 **<stdin>** 。

与 **ls** 不同，在缺省情况下， **lsacl** 不能自动列出子目录中文件的 ACL，也不能列出当前目录中文件的 ACL 条目。执行后一操作的较好方式是：

lsacl *

或者

lsacl .* *

外部语言环境影响

环境变量

LANG 用于确定显示消息的语言。

如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省的“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **lsacl** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

返回值

如果 **lsacl** 成功则返回零。如果调用错误，则返回值 **1** 。如果 **lsacl** 无法读取指定文件的 ACL ，则会在标准错误中输出一条错误消息，继续运行，并在稍后返回值 **2** 。

举例

列出文件 **dir/file1**的 ACL ：

lsacl dir/file1

以长格式列出当前目录中所有文件的 ACL。

lsacl -l .* *

列出 **mydir**下的所有文件的 ACL：

find mydir -print | sort | xargs lsacl

相关内容

如果目标文件驻留在一个不支持 ACL 的文件系统上， **lsacl** 将会失败。

NFS：

远程文件不支持 **lsacl** 。

作者

lsacl 由 HP 开发。

另请参阅

chacl(1)、 getaccess(1)、 ls(1)、 getacl(2)、 acl(5)。

名称

m4 - 宏处理器

概要

m4 [*options*] [*file* ...]

说明

m4 是一个可用作 Ratfor、C 和其他语言的前端的宏处理器。每个参数文件都按顺序进行处理；如果没有文件，或文件名为 **-** ，则从标准输入中读取。已处理的文本被写入标准输出中。

选项

m4 可识别下列选项:

-e 以交互模式进行操作。运行时会忽略中断，并且输出不会进入缓冲区。使用这种模式可能有很大的困难。

-s 启用 C 预处理器的行同步输出功能 (**#**line ...)

-B*int* 更改推回和参数收集缓冲区的大小的缺省值 4,096。

-H*int* 更改符号表散列数组大小的缺省值 199。该大小应为质数。

-S*int* 更改调用堆栈大小的缺省值 100 个插槽。宏占用 3 个插槽，非宏参数占用 1 个插槽。

-T*int* 更改令牌缓冲区大小的缺省值 512 字节。

上面列出的选项必须出现在任何文件名和任何 **-D** 或 **-U** 选项之前才能生效。

-D*name*[**=***val*]
将 *name* 定义为 *val* ，如果忽略 *val* ，则将名称定义为空。

-U*name* 取消定义 *name* 。

宏调用

宏调用的格式如下:

*name***(***arg1***,** *arg2***, ... ,***argn***)**

左括号 (() 必须直接跟在宏的名称之后。如果已定义的宏的名称后面没有 **(** ，则认为这是不带参数的宏调用。宏名称可能包括字母、数字和下划线 (_)；第一个字符不能为数字。

收集参数时，将忽略未用引号引起来的前导空格、制表符和换行符。左单引号和右单引号（**‘** 和 **’**）用于将字符串引起来。用引号引起来的字符串的值等于去掉引号的字符串。

当识别出宏名称后，可以通过搜索相匹配的右括号来收集宏的参数。如果收集到的宏参数比宏定义中的少，则末尾的参数被认为是空。宏的计算在参数收集过程中正常进行，并且对于出现在嵌套调用值中的任何逗号或右括号，其作用都与在原输入文本中的相同。收集完参数后，宏的值被推回到输入流中并被重新扫描。

内置宏名称

m4 可以有下列内置宏。这些宏可以被重新定义，但是一旦定义之后，其原来的含义就会丢失。除非另外规定，否则内置宏的值为空。

changecom 更改左右注释标记的缺省值 **#** 和换行符。没有参数时，会有效地禁用注释机制。有一个参数时，则左标记变成参数，而右标记变成换行符。有两个参数时，两个标记都会受到影响。注释标记最长为 5 个字符。

changequote 更改第一个和第二个参数的引用符号。这些符号最长为 5 个字符。不带参数的 **changequote** 将恢复原始值（例如，‘ 和 ’）。

decr 返回其参数值减 1 所得到的值。

define 将宏的名称用作第一个参数，将该宏的值设置为第二个参数。替换文本中出现的每个 **$***n*（其中 *n* 为数字）都被第 *n* 个参数替换。参数 0 为宏的名称；缺失的参数由空字符串替换；**$#** 由参数的数目替换； **$*** 由所有由逗号分隔的参数列表替换； **$@** 相当于 **$*** ，但是每个参数都用引号引起来（使用当前的引号）。

defn 返回其参数用引号引起来的定义。这对于重新命名宏是非常有用的，尤其是内置宏。

divert **m4** 可维护 10 个输出流，编号从 0 到 9。最后的输出是以数字顺序排列的彼此相串联的流；在最初，流 0 是当前的流。 **divert** 宏将当前的输出流更改为其（以数字字符串表示的）参数。转移到 0 到 9 以外的流的输出将被忽略。

divnum 返回当前输出流的值。

dnl 读取并忽略直到（并包括）下一个换行符之前的字符。

dumpdef 输出已命名项目的当前名称和定义，如果没有给出参数，则输出所有项目的当前名称和定义。

errprint 将其参数输出到诊断输出文件中。

eval 使用 32 位算法，将其参数作为算术表达式进行计算。运算符包括 **+** 、 **-** 、 ***** 、 **/** 、 **%** 、 ****** （求幂）， 位运算符 **&** 、 **|** 、 **^** 和 **~** , 关系运算符和括号。可以在 C 中指定八进制数和十六进制数。第二个参数可指定结果的基；缺省值为 10。第三个参数可以用来指定结果中的最小位数。

hpux 指预定义为空值的对象。

ifdef 如果已定义第一个参数，则其值为第二个参数；否则，其值为第三个参数。如果不存在第三个参数，则其值为空。词 **unix** 在 HP-UX 系统版本的 **m4** 中进行了预定义。

ifelse 有三个或更多的参数。如果第一个参数的字符串与第二个参数的相同，则其值为第三个参数。如果第一个参数的字符串与第二个参数的不相同，并且参数超过四个，则会对参数 4、5、6 和 7 重复此过程。否则，值或者为第四个字符串，或者（如果第四个字符串不存在）为空。

include 返回在参数中命名的文件的内容。

incr 返回其参数加 1 所得到的值。通过将最初的数字字符串转换为十进制数来计算参数的值。

index 在其第一个参数中返回第二个参数开始的位置（初始为零），如果第二个参数没有出现，则返回 -1。

len 返回其参数中的字符数。

m4exit 可直接从 **m4** 中退出。如果已给出，则参数 1 为退出代码；缺省代码为 0。

m4wrap 参数 1 被推回到最后的 EOF；例如： **m4wrap(‘cleanup()’)**

maketemp 用当前的进程 ID 填写其参数中的字符串 **XXXXX** 。

popdef 删除其参数的当前定义，并显示出参数以前的定义（如果有的话）。

pushdef 类似于 **define** ，但可保存任何以前的定义。

shift 返回除其第一个参数以外的所有参数。以逗号分隔的其他参数被引用并推回。该引用可使将在随后执行的重新扫描无效。

sinclude 类似于 **include** ，不同的是，它在文件不可访问时不作任何提示。

substr 返回其第一个参数的子字符串。第二个参数是选择第一个字符的原始为零的数；第三个参数表示子字符串的长度。如果第三个参数缺失，则认为它大到可延伸到第一个字符串的末尾。

syscmd 执行在第一个参数中给出的 HP-UX 系统命令。没有返回值。

sysval 指最后一次调用 **syscmd** 的返回代码。

traceoff 以全局方式为所有指定的宏关闭跟踪。只有明确地调用 **traceoff** ，才能不跟踪由 **traceon** 特别跟踪的宏。

traceon 没有参数时，启动跟踪所有的宏（包括内置宏）。否则，启动跟踪已命名的宏。

translit 将其第一个参数中的字符从第二个参数给定的集中转换到第三个参数给定的集。不允许缩写。

undefine 删除在其参数中命名的宏的定义。

undivert 可从命名为参数的转换中直接输出文本，如果没有参数，则从所有的转换中直接输出文本。文本可被取消转换到另一个转换中。取消转换会忽略已转换的文本。

（仅限于 XPG4。）指定包含内置宏的任何非数字字符的参数是错误的，这些内置宏有： **decr**、**"divert"** 、**incr**、**"m4exit"** 、 **substr**、**"undivert"** 和 **eval** 。

另请参阅

cpp(1)、 ratfor(1)。

符合的标准

m4：SVID2、SVID3、XPG2、XPG3、XPG4

名称

machid： hp9000s200、 hp9000s300、 hp9000s400、 hp9000s500、 hp9000s700、 hp9000s800、 hp-mc680x0、 hp-pa、 pdp11、 u3b、 u3b2、 u3b5、 u3b10、 u370、 vax - 提供有关处理器类型的真值

概要

hp9000s200

hp9000s300

hp9000s400

hp9000s500

hp9000s700

hp9000s800

hp-mc680x0

hp-pa

pdp11

u3b

u3b2

u3b5

u3b10

u370

vax

说明

如果处理器类型与命令名称匹配，则以下命令将返回值 true （退出代码为 0）。否则，将返回值 false（退出代码不为零）。这些命令通常在 **make** makefile 和 Shell 过程内使用，以改进应用程序的可移植性（请参阅 *make*(1)）。

命令	对于以下项为 **True**：	命令	对于以下项为 **True**：
hp9000s200	200 系列	***pdp11***	PDP-11/45 或 PDP-11/70
hp9000s300	300 系列	***u3b***	3B20 计算机
hp9000s400	400 系列	***u3b2***	3B2 计算机
hp9000s500	500 系列	***u3b5***	3B5 计算机
hp9000s700	700 系列	***u3b10***	3B10 计算机
hp9000s800	800/700 系列	***u370***	IBM System/370 计算机
hp-mc680x0	200、300 或 400 系列	***vax***	VAX-11/750 或 VAX-11/780
hp-pa	700 或 800 系列		

举例

假定有一个必须在 HP 9000 700 或 800 系列系统上运行时行为不同的 Shell 脚本，选择要执行的正确代码段：

```
if hp9000s800
then  # system is Series 700 or 800.
```

```
        if hp9000s700
        then  # System is Series 700

          # Series 700 code fragment goes here

        else   # System is Series 800

          # Series 800 code fragment goes here

        fi
      fi
```

警告

hp9000s800 在 800 系列和 700 系列系统上始终返回 true。因此，在脚本中使用此命令确定硬件类型时，始终按适当的顺序同时使用 **hp9000s800** 和 **hp9000s700** 以确保得到正确的结果（请参阅“举例”）。

machid(1) 将不再提供对 800 系列和 700 系列系统以后的将来计算机的支持。决策应该基于 *getconf*(1) 返回的硬件和软件配置信息。

另请参阅

getconf(1)、make(1)、sh(1)、test(1)、true(1)。

名称

machinfo - 输出计算机信息

概要

machinfo

说明

machinfo 输出与计算机有关的有用调试信息。该信息包括芯片步进、固件版本、 CPU 数量和内存容量。

请注意 CPU 特有的数据将只适用于运行 **machinfo** 命令的处理器。在其他 CPU 上运行 **machinfo** 命令可能会导致显示不同的 CPU 特有信息。使用 *mpsched*(1) 命令可在其他处理器上强制执行 **machinfo** 。

另请参阅

getconf(1)、 mpsched(1)、 uname(1)。

名称

mail、rmail - 向用户发送邮件或读取邮件

概要

mail [**+**] [**-epqr**] [**-f** *file*]

mail [**-dt**] *person* ...

rmail [**-dt**] *person* ...

备注:

有关增强的用户邮件接口，请参阅 *mailx*(1) 和 *elm*(1) 。

说明

在没有参数的情况下使用 **mail** 命令时，该命令按后进先出的顺序逐个打印用户的邮件。

对于每封邮件， **mail** 打印 **?** 提示符并从标准输入读取行，以确定邮件的处理方法。如果 **mail** 已经到达最后一封邮件，则自动前进到下一封邮件的命令将从 **mail** 退出。

命令

mail 支持以下命令:

<换行符>	继续到下一封邮件。如果已到达最后一封邮件，则退出。
+	与 <换行符> 相同。
n	与 <换行符> 相同。
d	删除该邮件并继续到下一封邮件。
p	再次打印邮件。
-	返回到上一封邮件。
s [*files*]	在指定 *files* （缺省值为 **mbox** ）中保存邮件，将邮件标记为要从用户的 *mailfile* 中删除，并继续到下一封邮件。
y [*files*]	与 **s** [*files*] 相同。
w [*files*]	在指定的 *files* （缺省值为 **mbox** ）中保存邮件但不保存其标头（“发件人...”行），将该邮件标记为要删除，然后继续到下一封邮件。
m *person* ...	将邮件发送给每个指定 *person* ，将邮件标记为要删除，然后继续到下一封邮件。
q	将未删除的邮件放回 **mailfile** 中并停止。
EOT (Ctrl-D)	与 **q** 相同。
x	异常中止。保持原始 **mailfile** 不变并停止。
!*command*	退出到命令解释程序并执行 *command* 。

? 打印命令摘要。

***** 与 **?** 相同。

命令行选项

以下命令行选项用于更改邮件的打印方式:

+ 使邮件按先进先出顺序打印。

-e 禁止打印邮件并返回退出值:

0 = 邮件存在
1 = 无邮件
2 = 其他错误

-p 打印所有邮件，而不提示您如何处理。

-q 如果收到中断信号，则会导致 **mail** 终止。通常，中断信号仅导致当前邮件的打印终止。

-r 与 **+** 相同。

-f *file* 导致 **mail** 使用 *file* （例如， **mbox** ）而不是缺省的 *mailfile* 。

-t 导致出站邮件的前面有要向其发送邮件的每个 *person* 。 *person* 通常为 **login** 可识别的用户名（请参阅 *login*(1) ）。如果无法识别向其发送邮件的 *person* ，或者如果在输入期间中断了 **mail** ，则将保存文件 **dead.letter** 以允许编辑和重新发送。请注意， **dead.letter** 被视为临时文件，因为在每次需要时重新创建它，并删除 **dead.letter** 的以前内容。

-d 导致 **mail** 直接传递邮件。这样就将 **mail** 与作出路由决定隔离开，并允许将它用作本地传递代理。通常，自动路由工具在本地传递邮件时使用此选项。

如果指定了 *person* ， **mail** 接收直到文件结束标志的标准输入（或者直到仅包含 **.** 的行），并将它添加到每个 *person* 的 *mailfile* 。邮件的前面是发件人的姓名和邮戳。

要表示远程系统上的收件人，请在 *person* 前面添加系统名和感叹号（请参阅 *uucp*(1) ）。 *person* 中第一个感叹号之后的所有内容都由远程系统加以解释。具体说来，如果 *person* 包含其他感叹号，则它可以表示将邮件发送到其最终目标途经的计算机序列。例如，如果将 **a!b!cde** 指定为收件人的名称，则导致将邮件发送到系统 **a** 上的用户 **b!cde** 。然后系统 **a** 将该目标解释为请求，以将邮件发送到系统 **b** 上的用户 **cde** 。这可能是很有用的，例如，如果发送系统可以访问系统 **a** ，但是不能访问系统 **b** 。如果远程系统是本地系统名（即 localsystem!user），则 **mail** 不使用 **uucp** 。

可以通过两种方法处理 **mailfile** 以更改 **mail** 的功能。文件的 其他权限可以为读写、只读或既不可读也不可写，以允许其他保密级别。如果更改为除缺省值之外的值，则将保留文件（即使它为空）以便使所需权限永久存在。该文件也可以包含第一行:

Forward to *person*

导致将发送到 **mailfile** 所有者的所有邮件转发到 *person* 。对于在多计算机环境中将某个用户的所有邮件转发到给

定的计算机，这尤其有用。为了使转发正常工作， **mailfile** 应将“mail”作为组 ID ，而且组权限应该是读写。

rmail 仅允许发送邮件。 **uucp** 将 **rmail** 用作安全预防措施。

在用户登录时，可以使用命令 **mail -e** 检测是否存在邮件（如果有的话），并这样指示。在终止时，如果新邮件已到但 **mail** 正在运行，则 **mail** 生成通知消息。

外部语言环境影响

环境变量

LC_TIME 用于确定所显示日期和时间字符串的格式和内容。

如果未在环境中指定 **LC_TIME** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果未指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **mail** 将认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

如果设置了 **TMPDIR** 环境变量，则它指定要用于临时文件的目录，并覆盖缺省目录 **/tmp** 。

国际代码集支持

在 HP-UX 系统之间，在邮件文本内支持单字节字符和多字节字符代码集。在标头中只能使用 7 位 USASCII 代码集中的字符（请参阅 *ascii*(5) ）。

警告

有时可能导致无法删除锁定文件的错误。

在中断后，可能不打印下一封邮件。要强制打印，请键入 **p** 。

类似邮件中邮戳的行（即“发件人 ...”）的前面有 **>** 。

mail 将仅由点 (**.**) 组成的行视为邮件的结尾（使用 **rmail -d** 命令时除外）。

在邮件中允许的最大行长度为在 **/usr/include/stdio.h** 中定义的 **BUFSIZ** 字节数的 8 倍。如果行长度超过此限制，则 **mail** 将截断从行开始标志处开始的行，并仅使用尾随的 8 * **BUFSIZ** 个字符。

使用两个单独的邮件程序同时访问同一邮件文件（通常是无意中来自两个单独的窗口）可以导致不可预知的结果。

如果缺少下面的任何收件人字段，则具有严格遵循 RFC-822 的程序的一些站点将无法传递邮件。

To:
resent-to:
cc:
resent-cc:
bcc:
resent-bcc:

您可以将 RFC-822 命令添加到邮件程序缓冲区（或编辑器）。例如：

mail user1@domain.com
From: user2@domain.com

Subject: This is a test
To: test_address@domain.com

This is a test

文件

/var/mail/*.lock	用于邮件目录的锁
dead.letter	无法发送的文本
/tmp/ma*	临时文件
$MAIL	包含 **mailfile** 的路径名的变量
$HOME/mbox	已保存的邮件
/etc/passwd	标识发件人和查找人员
/var/mail	收到的邮件的目录（模式 **775** ，组 ID **mail** ）
/var/mail/*user*	*user* 收到的邮件；即 **mailfile** （模式 **660** ，组 ID **mail** ）

另请参阅

login(1)、 mailx(1)、 uucp(1)、 write(1)。

符合的标准

mail： SVID2、SVID3、XPG2、XPG3

rmail： SVID2、SVID3

名称

mailfrom - 按主题和发件人汇总邮件文件夹

概要

mailfrom [**-hnQqStv**] [**-s** *status*] [*folder*|*username*]...

说明

mailfrom 命令读取一个或多个邮件文件夹，并为每封邮件输出一行内容，格式为：

from [*subject*]

其中， *from* 是邮件发件人的姓名， *subject* 是邮件的主题（如果有）。如果 **mailfrom** 确定邮件是您发送的，则 *from* 部分将显示 **To** *user* ，其中 *user* 是邮件发送到的用户。当您收到自己发送的邮件时，就会发生这种情况。

缺省文件夹是收件箱 **/var/mail/***yourloginname* 。请参阅下面的“操作数”小节。

选项

mailfrom 可识别下列选项：

-h 输出概述选项的简短帮助消息。

-n 使用 **readmail** 所用的同一编号方案对邮件进行编号。

-Q 完全静态模式。仅生成错误消息。该选项在 Shell 脚本中很有用，在该脚本中只有程序的成功或失败是重要的，并不需要输出。

-q 静态模式。对于每个邮箱或文件夹，仅输出一行摘要信息。

-S 在每个邮箱或文件夹中，按邮件状态添加一条邮件数量摘要信息。要仅获取摘要，请与 **-q** 选项一起使用。摘要格式如下：

Folder contains:
New messages: *n*
Unread messages: *u*
Read messages: *r*

如果某一项计数 *n* 、 *r* 或 *u* 为零，该行将被忽略。

-s *status* 仅显示具有给定状态的邮件的标题。 *status* 可以是 **new** 、 **old** 、 **read** 或 **unread** 其中之一。 **old** 和 **unread** 是等效的。可以重复使用 **-s** 选项，以便从多个类别输出标题信息，例如只有新邮件和未读邮件。值可以简写为它们的首字母。缺省值是所有邮件。

-t 整理模式。如果 *from* 字段足够长，可以从普通起始列开始移置 *subject* 字段，则会将主题向下移动到下一行。

-v 详细模式。在列出每个邮箱或文件夹的内容之前，输出一个描述性标题。

操作数

mailfrom 可识别下列可选的操作数:

folder|username 文件名或系统上邮件用户的名称。可以使用 **=***filename* 格式指定邮件目录中的文件夹，该目录是由 **elmrc** 配置文件中的 **maildir** 字符串变量定义的。

mailfrom 将搜索文件名形式的相对于当前目录的值。然后，如果该文件名不是绝对路径，它将搜索相对于收件箱目录 **/var/mail** 的值。找到的第一个文件将被选中。您必须对该文件有读取权限。

退出状态

mailfrom 将返回下列值:

0 出现与 *status* 匹配的邮件。

1 没有出现与 *status* 匹配的邮件，但是存在一些邮件。

2 根本没有任何邮件。

3 发生了错误。

如果指定了多个邮箱或文件夹，则退出状态仅适用于最后一个检查的邮件。这可以在脚本中使用，以确定用户具有何种邮件。

举例

显示邮箱中所有邮件的标题信息。

mailfrom

显示邮箱中所有新邮件的标题信息。

mailfrom -s new

假定您有读取 **guest** 的邮件的适当文件权限，输出 **guest** 的收件箱中所有新邮件和未读邮件的标题信息。

mailfrom -s new -s unread guest

仅输出收件箱中新邮件、未读邮件和已读邮件数量的摘要信息。

mailfrom -q -S

文件

$HOME/.elm/elmrc **elm** 配置文件。

/var/mail 收件箱的目录。

作者

mailfrom 由 HP 开发。

另请参阅

elm(1)、 mail(1)、 mailx(1)、 readmail(1)。

名称

mailq - 输出邮件队列概要

概要

mailq [**-v**]

说明

mailq 可输出队列中将来要发送的邮件消息的概要。

为每条消息输出的第一行显示了该主机上使用的消息内部标识符、以字节表示的消息大小、队列接收该消息的日期和时间以及消息发送人。第二行显示了致使该消息滞留在队列中的错误消息；第一次处理消息时，不会显示这一错误消息。状态字符为：

***** 表示正在处理该作业

X 表示负载过大以致无法处理该作业

- 表示作业在列队中很新，尚未轮到该作业。

第二行之后的每一行显示一个消息接收人。

mailq 与 **sendmail -bp** 等同。

该命令支持下列选项：

选项

-v 输出详细信息。该选项还可显示消息的优先级，以及一个单字符指示器（“+”或空），该字符可表明是否在消息的第一行发送了警告消息。另外，还可在接收人部分混合显示其他行，来表明“控制用户”信息。这种信息给出了谁是为该消息执行的程序的所有者，以及该命令对应的别名（如果有的话）。

返回值

mailq 实用程序成功时返回 0，发生错误时返回 > 0 的值。

作者

mailq 由加州大学伯克利分校开发，最开始出现在 4.0BSD 中。

文件

/var/spool/mqueue/* 邮件列队文件

另请参阅

sendmail(1M)。

名称

mailstats - 输出邮件流量统计信息

概要

mailstats [**-C** *cffile*] [**-f** *stfile*] [**-o**] [**-p**]

说明

mailstats 读取和解释 **sendmail** 统计文件，然后输出邮件流量统计信息。统计文件是由 **/etc/mail/sendmail.cf** 中的 **StatusFile** 选项设置的。缺省的统计文件是 **/etc/mail/sendmail.st** 。如果统计文件存在，则 **sendmail** 会收集有关邮件流量的统计信息，并将它们存储在该文件中。该文件不会增长。

对于 **sendmail** 配置文件中定义的每个邮件程序，统计信息是以每个邮件程序为基础进行收集的。对于所有入站和出站流量，统计信息是以邮件数和字节数的形式进行保存的。

mailstats 实用程序显示在第一行开始收集统计信息的时间。然后，每个邮件程序的统计信息显示在单独的一行上，每一行都带有下列以空格分隔的字段（请参阅下面的“举例”一节）：

M 邮件程序编号。

msgsfr 来自邮件程序的邮件数量。

bytes_from 来自邮件程序的千字节数。

msgsto 发给邮件程序的邮件数量。

bytes_to 发给邮件程序的千字节数。

msgsrej 被拒绝的邮件数量。

msgsdis 被忽略的邮件数量。

Mailer 邮件程序的名称。

在显示这些内容之后，将显示一个包含所有邮件程序各个值的总计信息的行，用一个仅包含等号 (“=”) 字符的行将其与前面的信息进行分隔。

注释

只有有权限的用户才能使用 **mailstats** 。

选项

选项如下:

-C *cffile* 读取指定的 *cffile* ，而不是缺省的 **/etc/mail/sendmail.cf** 文件。

-f *stfile* 读取指定的统计文件 *stfile* ，而不是在 **/etc/mail/sendmail.cf** 文件中指定的统计文件。

-o 不显示输出中邮件程序的名称。

-p 在“可读程序”模式下输出信息，并清除统计信息。

要清除统计文件，请以超级用户身份执行以下命令:

cp /dev/null *statistics-file*

返回值

mailstats 实用程序成功时返回 0，发生错误时返回 >0 的值。

诊断信息

如果统计文件不可访问，或者更改了统计文件的大小，则 **mailstats** 会生成错误消息。错误消息如下：

mailstats: file size changed

statistics-file 长度为零（表示自清除统计文件后没有传输任何邮件），或者统计文件的大小已经更改。由于该文件的大小应该是保持不变的，因此大小的任何变化都意味着该文件无效。

mailstats: *statistics-file***: No such file or directory**

统计文件不存在。

mailstats: *statistics-file***: Permission denied**

设置了统计文件的权限，因此您无法读取。

举例

下面是 **mailstats** 输出的典型示例：

Statistics from Thu Jul 11 14:12:40 1996

M	msgsfr	bytes_from	msgsto	bytes_to	msgsrej	msgsdis	Mailer
0	0	0K	3	4K	0	0	prog
3	3	4K	0	0K	0	0	local
5	2	1K	11	11K	4	0	esmtp
============							
T	13	13K	14	15K	4	0	

该示例显示了邮件程序 0、3 和 5 自 7 月 11 日星期四以来已经处理了给定数量的邮件流量。具体地说， **sendmail** 通过邮件程序 esmtp (M 5) 已经发送了包含 11 KB 的 11 个邮件，但通过邮件程序 esmtp (M 5) 有 4 个邮件被拒绝。

作者

mailstats 由加州大学伯克利分校开发。

文件

/etc/mail/sendmail.st	缺省的邮件流量统计文件
/etc/mail/sendmail.cf	sendmail 配置文件

另请参阅

sendmail(1M)。

名称

mailx - 交互式邮件消息处理系统

概要

发送模式

mailx [**-FUm**] [**-s** *subject*] [**-r** *address*] [**-h** *number*] *address* ...

接收模式

mailx -e

mailx [**-UHLiNn**] [**-u** *user*]

mailx -f [**-UHLiNn**] [*filename*]

过时形式

mailx [**-f** *filename*] [**-UHLiNn**]

说明

mailx 为发送和接收电子邮件提供了一个舒适灵活的环境。读取邮件时， **mailx** 提供了一些命令，用以帮助保存、删除和回复邮件。发送邮件时， **mailx** 可用于对邮件进行编辑、查看和其他修改，就像创建邮件一样。

每个用户收到的邮件保存在该用户的名为 **system mailbox** 的标准文件中。使用 **mailx** 读取邮件时，将使用该系统邮箱，但使用带特定文件名或不带特定文件名的 **-f** 选项指定一个备用邮箱文件时除外。从系统邮箱读取传入邮件时，邮件会被标记为移动并保存到备用文件（除非采用特定操作），以便不必再见到这些邮件。此备用文件称为 *mbox* ，通常位于用户的 **HOME** 目录中（有关此文件和 **mailx** 使用的其他环境变量的说明，请参阅环境变量小节中的 **MBOX**）。邮件会一直保留在此文件中，直到被明确删除。

命令行选项以连字符 (**-**) 开头，并假定任何其他参数为目的地（收件人）。

必须用引号将包含多个词的参数括起来。

如果未指定收件人，则 **mailx** 将尝试从系统邮箱读取邮件。

命令行上指定的收件人地址的总长度必须小于 1024 字符。您可以声明一个 **alias** 或 **group** （请参阅命令一节），以指定长达 8191 字符的收件人地址或地址列表，并使用该别名或组名（但是列表中的每个地址仍必须小于 1024 字符）。如果希望指定大于此长度的收件人地址列表，请让系统管理员在系统别名文件 **/etc/mail/aliases** 中声明一个别名或组，然后使用该别名名称。

选项

mailx 可识别下列命令行选项：

-e 测试是否有邮件。如果有要读取的邮件，则 **mailx** 不输出任何内容，并在退出时返回成功返回代码。有时，用于登录脚本（如 **$HOME/.profile** ），以便在登录时检查邮件。

-f 从 *filename* 而不是用户的系统邮箱中读取邮件。如果未指定 *filename* ，则使用备用 *mbox* 。

-F 将邮件记录在以第一个收件人命名的文件中。覆盖 **record** 环境变量（如果已设置）。

-h *number* 目前已完成的网络“跳跃”数。为网络软件提供此选项以防止无限发送循环。

-H 仅输出标题摘要。

-L 仅输出完整的标题信息。

-i 忽略中断。另请参阅下文的 **ignore** 环境说明。

-n 不要从系统缺省的 **mailx.rc** 文件进行初始化。

-m 请勿在发送邮件时向标题信息添加 MIME 标题行 *Mime* 版本、内容类型和 内容编码。

-N 不输出初始标题摘要。

-r *address* 将 *address* 传递给网络发送软件。禁用所有波形符命令。

-s *subject* 将“主题”标题字段设置为 *subject* 。

-u *user* 读取 *user* 的 *mailbox* 。只有当 *user* 邮箱的读取权限没有受到读取保护时，才可使用。

-U 将 UUCP 样式的地址转换为 Internet 标准。覆盖 **conv** 环境变量。

-d 打开调试输出。不是特别重要，不推荐使用。

读取邮件时， **mailx** 在 命令模式中运行。会显示前几个邮件的标题摘要，接着出现提示，指出 **mailx** 可以接受常规命令（请参阅命令一节）。发送邮件时， **mailx** 在 输入模式中运行。如果命令行上未指定主题，则输出提示，以提示输入主题。键入邮件时， **mailx** 会读取邮件并将其保存到临时文件中。通过换行并以波形符 (˜) 转义字符开头，后接单个命令字母和可选参数，可以输入命令。有关这些命令的摘要，请参阅波形符转义字符一节。

mailx 在任意指定时间的行为由一组 环境变量进行管理；这组环境变量是使用 **set** 和 **unset** 命令设置及清除的标志和已赋值参数。有关这些参数的摘要，请参阅环境变量小节。

命令行上列出的收件人可以是这三种类型：登录名、Shell 命令或别名组。登录名可以是任何网络地址，包括混合网络地址。如果收件人的名称以管道符号 (|) 开头，则假定名称的剩余部分为 Shell 命令，并通过管道传输邮件。这就为读取标准输入的任何程序提供了一个自动接口，例如， **lp** （请参阅 *lp*(1) ）可将外发的邮件记录在纸上。使用 **alias** 命令（请参阅命令一节）可以设置别名组，别名组是任何类型收件人的列表。

常规命令使用以下格式

[*command*] [*msglist*] [*arguments*]

如果在命令模式中未指定任何命令，则假定为 **print** 。在输入模式中，通过转义字符（除非用 **escape** 环境变量重新定义，否则为波形符）识别命令，不作为命令处理的行视作邮件的输入。

会为每个邮件分配一个序号，并始终对 当前邮件加以指明，在标题摘要中用 **>** 对其进行标记。许多命令使用可选邮件列表 (*msglist*) 进行操作，缺省列表为当前邮件。 *msglist* 是用空格分隔的邮件规范列表。邮件列表可以包含：

n 邮件编号 *n* 。

. 当前邮件。

^ 第一个未删除邮件。

$ 最后一个邮件。

***** 所有邮件。

*n***-***m* 邮件编号的范围（含边界），从 *n* 至 *m* ，其中 *n* 小于 *m* 。

user 来自 *user* 的所有邮件。

/*string* 主题行中包含 *string* （忽略大小写区分）的所有邮件。

:*c* 类型为 *c* 的所有邮件，其中 *c* 为下列值之一：

- **d** 已删除邮件
- **n** 新邮件
- **o** 旧邮件
- **r** 已读邮件
- **u** 未读邮件

请注意，命令的上下文决定了这种类型的邮件规范是否有意义。

其他参数通常是任意字符串，字符串的用法取决于涉及的命令。

需要时，文件名可使用普通的 Shell 约定进行扩展（请参阅 *sh*(1)）。某些命令可识别特殊的字符，这些字符将在下文中与命令一起进行说明。

启动时， **mailx** 从系统级文件 (**/usr/share/lib/mailx.rc**) 中读取命令，以初始化某些参数，然后从专用启动文件 (**$HOME/.mailrc**) 中初始化个性化变量。大多数常规命令在启动文件中是合法的，最常用的用途是设置初始显示选项和别名列表。但下面的命令在启动文件中不合法： **!** 、 **C**opy 、 **e**dit 、 **fo**llowup 、 **F**ollowup 、 **ho**ld 、 **m**ail、 **pre**serve 、 **r**eply 、 **R**eply 、 **sh**ell 和 **v**isual 。启动文件中的任何错误都会导致该文件中的剩余行被忽略。

命令

mailx 命令的完整列表如下：

! *command* 转义至 Shell。请参阅下文的 **SHELL** 环境变量说明。

*comment* Null（空）命令（注释）。在 **.mailrc** 文件中很有用。

= 输出当前邮件编号。

? 输出命令摘要。

newline 前进到下一个邮件，并使用 **print** 输出内容。如果这是输入的第一个命令，则输出第一个未读邮件。（要阅读当前邮件，请使用 **print** ）。

alias *alias name*...
group *alias name*...
为给定名称声明别名。当使用 *alias* 作为收件人时，将替换相应名称。在 **.mailrc** 文件中很有用。

alternates *name*...
声明一个备用登录名列表。响应邮件时，会将这些名称从该响应的收件人列表中删除。如果不带参数，则 **alt**ernates 会输出当前的备用名列表。另请参阅环境变量小节中的 **allnet**。

cd [*directory*]
chdir [*directory*]
更改目录。如果未指定 *directory* ，则使用 **$HOME** 。

copy [*filename*]
copy [*msglist*] *filename*
将邮件复制到文件，但邮件不标记为已保存。在其他方面等效于 **s**ave 命令。

Copy [*msglist*]
将指定邮件保存到文件（其名称从要保存的邮件的作者派生而来）中，但邮件不标记为已保存。在其他方面等效于 **S**ave 命令。

delete [*msglist*]
从 *mailbox* 中删除邮件。如果已设置 **autoprint** ，则输出最后一个已删除邮件之后的下一个邮件（请参阅环境变量小节）。另请参阅 **dp** 。

discard [*header-field* ...]
ignore [*header-field* ...]
在屏幕上显示邮件时，不输出指定的标题字段。可以忽略的标题字段示例包括“状态”和“抄送”。保存邮件时，则包含这些字段。 **P**rint 和 **T**ype 命令会覆盖此命令。

dp[*msglist*]
dt[*msglist*]
从邮箱中删除指定邮件，并输出最后一个已删除邮件之后的下一个邮件。大致等效于 **d**elete 命令后面紧接 **p**rint 命令。

echo *string* ...
回显一个或多个给定的字符串（类似于 **echo** ，请参阅 *echo*(1) ）。

edit [*msglist*]
编辑给定的邮件。邮件位于临时文件中，可使用 **EDITOR** 变量获取编辑器的名称（请参阅环境变量小节）。缺省的编辑器为 *ed* （请参阅 *ed*(1) ）。

exit
xit
退出 **mailx** ，且不更改邮箱。不在 *mbox* 中保存邮件（另请参阅 **q**uit ）。

file [*filename*]
folder [*filename*]
退出当前的邮件文件并在指定文件中读取。当以下几个特殊字符用作文件名时，可以进行识别，替换方式如下:

%	当前邮箱。
%*user*	*user* 的邮箱。
#	前一个文件。
&	当前 *mbox* 。

缺省文件是当前邮箱。

folders 输出目录中文件的名称，该目录由 **folder** 变量设置（请参阅环境变量小节）。

followup [*message*] 对一个邮件作出响应，并将响应记录到文件（其名称从该邮件的作者派生而来）中。覆盖 **record** 变量（如果已设置）。另请参阅 **F**ollowup 、 **S**ave 和 **C**opy 命令以及 **outfolder** （请参阅环境变量小节）。

Followup [*msglist*] 对 *msglist* 中的第一个邮件作出响应，将该邮件发送至 *msglist* 中每个邮件的作者。将从第一个邮件中提取主题行，同时将响应记录到文件（其名称从第一个邮件的作者派生而来）中。另请参阅 **fo**llowup 、 **S**ave 和 **C**opy 命令以及 **outfolder** （请参阅环境变量小节）。

from [*msglist*] 输出指定邮件的标题摘要。

group *alias name...*
alias *alias name...* 为给定名称声明别名。当使用 *alias* 作为收件人时，将替换相应名称。在 **.mailrc** 文件中很有用。

headers [*message*] 输出包含指定邮件的标题页。 **screen** 变量设置每页的标题数（请参阅环境变量小节）。另请参阅 **z** 命令。

help 输出命令的摘要。

hold [*msglist*]
preserve [*msglist*] 将指定邮件保存在 *mailbox* 中。

if **s**|**r**
  *mail-command*s
else
  *mail-command*s
endif 条件执行，其中 **s** 执行附带的 *mail-command* ，直到 **el**se 或 **en**dif （如果程序处于发送模式）； **r** 使附带的 *mail-command* 仅在接收模式中执行。用于 **.mailrc** 文件中。

ignore *header-field ...*
discard *header-field ...*
在屏幕上显示邮件时，不输出指定的标题字段。可以忽略的标题字段示例包括 **status** 和 **cc**。保存邮件时，则包含所有字段。 **P**rint 和 **T**ype 命令会覆盖此命令。

list 输出所有可用的命令。不给出说明。

mail *name ...* 将邮件发送给指定用户。

mbox [*msglist*] 当 **mailx** 正常终止时，安排给定邮件在标准 *mbox* 保存文件中结束。有关此文件的说明，请参阅环境变量小节中的 **MBOX** 。另请参阅 **ex**it 和 **q**uit 命令。

next [*message*] 转至与 *message* 相匹配的下一个邮件。可指定一个 *msglist* ，但这种情况下，唯一使用的邮件就是列表中第一个有效邮件。这对于从特定用户跳转到下一个邮件很有用，因为在不给

出实际命令时，会将名称解释为命令。有关可能的邮件规范的说明，请参阅上文的 *msglist* 论述。

pipe [*msglist*] [*command*]

| [*msglist*] [*command*] 将 *msglist* 中的邮件通过管道传送给指定 *command* 。将处理每个邮件，就像读取邮件一样。如果未指定 *msglist* ，则使用当前邮件。如果未指定 *command* ，则使用由 **cmd** 变量的当前值指定的命令。如果指定了 *msglist* ，则也必须指定 *command* 。如果设置了 **page** 变量，则每个邮件的后面都会插入一个换页符（请参阅环境变量小节）。

preserve [*msglist*]

hold [*msglist*] 保留 *mailbox* 中的指定邮件。

Print [*msglist*]

Type [*msglist*] 在屏幕上输出指定邮件，包含所有标题字段。覆盖 **ig**nore 命令行使的禁止显示字段功能。

print [*msglist*]

type [*msglist*] 输出指定的邮件。如果设置了 **crt** ，则邮件长度长于 **crt** 变量指定的行数时，将通过 **PAGER** 变量指定的命令进行分页。缺省命令是 **pg** （请参阅 *pg*(1) ），但许多用户偏爱 **more** （请参阅 *more*(1) ；参阅环境变量小节）。

quit 退出 **mailx** ，同时保存 *mbox* 中的已读邮件以及用户系统邮箱中的未读邮件。已明确地保存到某个文件中的邮件将被删除。

Reply [*msglist*]

Respond [*msglist*] 向 *msglist* 中的每个邮件的作者发送响应。将从第一个邮件中提取主题行。如果 **record** 已设置为文件名，则将响应保存到该文件的结尾（请参阅环境变量小节）。

reply [*message*]

respond [*message*] 回复指定的邮件，包含该邮件的所有其他收件人。如果 **record** 已设置为文件名，则将响应保存到该文件的结尾（请参阅环境变量小节）。

Save [*msglist*] 将指定的邮件保存到文件（其名称从第一个邮件的作者派生而来）中。文件名基于作者的名称，并去除所有网络地址。另请参阅 **C**opy 、 **fo**llowup 和 **F**ollowup 命令以及 **outfolder** （请参阅环境变量小节）。

save [*filename*]

save [*msglist*] *filename*

将指定的邮件保存到给定文件中。如果文件不存在，则创建该文件。当 **mailx** 终止时，邮件将从 *mailbox* 中删除，除非设置了 **keepsave** （请参阅环境变量小节以及 **exit** 和 **q**uit 命令）。

set

set *name*

set *name*=*string*

set *name=number* 定义一个名为 *name* 的变量。变量可被赋予 null（空）、字符串或数字值。使用 **Set** 自身会输出所有已定义的变量及其值（有关 **mailx** 变量的详细说明，请参阅环境变量小节）。

shell 调用交互式 Shell（请参阅环境变量小节中的 **SHELL** ）。

size [*msglist*] 输出指定邮件的大小（以字符数计）。

source *filename* 从给定文件中读取命令并返回到命令模式。

top [*msglist*] 输出指定邮件的开始几行。如果设置了 **toplines** 变量，则将其解释为要输出的行数（请参阅环境变量小节）。缺省值为 5。

touch [*msglist*] 处理指定邮件。如果 *msglist* 中的任何邮件没有专门保存到文件中，则在正常终止时，该邮件会被放置到 *mbox* 中。请参阅 **ex**it 和 **q**uit 。

Type [*msglist*]
Print [*msglist*] 在屏幕上输出指定邮件，包含所有标题字段。覆盖 **ig**nore 命令行使的禁止显示字段功能。

type [*msglist*]
print [*msglist*] 输出指定的邮件。如果设置了 **crt** ，则邮件长度长于 **crt** 变量指定的行数时，将通过 **PAGER** 变量指定的命令进行分页。缺省命令是 *pg*(1) ，但许多用户偏爱 *more*(1) （请参阅环境变量小节）。

unalias *alias* 忽略指定的 *alias* 名称。

undelete [*msglist*] 恢复指定的已删除邮件。仅恢复当前邮件会话中已被删除的邮件。如果设置了 **autoprint** ，则输出这些被恢复邮件中的最后一个邮件（请参阅环境变量小节）。

unset *name***...** 可清除指定变量。如果变量是从执行环境中导入的 Shell 变量，则无法清除该变量。

version 输出当前版本和发行日期。

visual [*msglist*] 使用屏幕编辑器编辑给定的邮件。邮件将被放置到临时文件中，可以使用 **VISUAL** 变量获取编辑器的名称（请参阅环境变量小节）。

write [*msglist*] *filename*
将给定邮件写入指定文件，标题（“发件人 ...”行）和结尾空行除外。在其他方面等效于 **s**ave 命令。

xit
exit 退出 **mailx** ，且不更改 *mailbox* 。不在 *mbox* 中保存邮件（另请参阅 **q**uit ）。

z[**+**|**-**] 将标题显示向前或向后滚动一个满屏。显示的标题数由 **screen** 变量设置（请参阅环境变量小节）。

波形符转义字符

以下命令仅可在输入模式下使用，使用方式是在命令行前加波形符转义符 (~) 。有关更改此特殊字符的信息，请参阅环境变量小节中的 **escape** 。

~!*command* 转义至 Shell。

~. 模拟文件结束（终止邮件输入）。

~:*mail-command*
~_ *mail-command* 执行命令级请求。仅当发送邮件和读取邮件两者同时进行时才有效。

~? 输出波形符转义字符的摘要。

~A 将签名字符串 **Sign** 插入邮件中（请参阅环境变量小节）。

~a 将签名字符串 **sign** 插入邮件中（请参阅环境变量小节）。

~b *name ...* 将 *name* 添加到密件抄送 (Bcc) 列表。

~c *name ...* 将 *name* 添加到抄送 (Cc) 列表。

~d 在 **dead.letter** 文件中读取。有关此文件的说明，请参阅环境变量小节中的 **DEAD** 。

~e 对部分邮件调用编辑器。另请参阅下文的 **EDITOR** 环境变量说明。

~f [*msglist*] 转发指定的邮件。指定的邮件将插入到转发邮件中，没有任何变化。

~h 提示输入主题行以及收件人、抄送和密件抄送列表。如果字段显示有初始值，则可对其进行编辑，就像是刚键入的一样。

~i *string* 将已命名变量的值插入到邮件文本中。例如， **~A** 等效于 **~i Sign** 。

~m [*msglist*] 将指定邮件插入邮件文本中，并将新文本向右移动一个制表位。仅当读取邮件的同时又发送邮件时才有效。

~p 输出正在输入的邮件。

~q 通过模拟中断退出（终止）输入模式。如果邮件的正文非 null（空），则部分邮件会保存在 **dead.letter** 中。有关此文件的说明，请参阅下文的 **DEAD** 环境变量说明。

~R *name ...* 将 *name* 添加到回复人列表。

~r *filename*
~< *filename*
~<!*command* 在指定文件中读取。如果参数以感叹号 (**!**) 开头，则假定剩余字符串为任意 Shell 命令并执行它，同时将标准输出插入到邮件中。

~s *string ...* 将主题行设置为 *string* 。

~t *name ...* 将给定 *name* 添加到收件人列表。

~v 对部分邮件调用首选屏幕编辑器。另请参阅下文的 **VISUAL** 环境变量说明。

~w *filename* 将部分邮件写入到指定文件中，但不包含标题。

~x 与使用 **~q** 一样地退出，但不将邮件保存到 *dead.letter* 中。

~| *command* 将邮件正文通过管道传送至给定 *command* 。如果 *command* 返回成功退出状态，则命令的输出将替换该邮件。

外部语言环境影响

环境变量

下列变量为内部 **mailx** 程序变量。任何时候都可从执行环境中导入这些变量，或者使用 **set** 命令设置它们。 **uns**et 命令可用于清除变量。

allnet 登录名相互匹配的所有网络名都视为相同。这使 *msglist* 的邮件规范也有类似行为。缺省值是 **noallnet** 。另请参阅 **alt**ernates 命令和 **metoo** 变量。

append 终止时，将邮件追加到 *mbox* 文件的末尾，而不是插入到该文件的开头。缺省值是 **noappend** 。

askbcc 输入邮件后提示输入密件抄送列表。缺省值是 **noaskbcc** 。

askcc 输入邮件后提示输入抄送列表。缺省值是 **noaskcc** 。

asksub 如果没有使用 **-s** 选项在命令行上指定主题，则提示输入主题。缺省情况下启用。

autoprint 启用在执行 **d**elete 和 **u**ndelete 命令之后自动输出邮件。缺省值是 **noautoprint** 。

bang 在 Shell 转义命令行中启用感叹号 (**!**) 的特例处理，与 *vi*(1) 中相同。缺省值是 **nobang** 。

charset=*charset* 设置缺省字符集。如果未指定缺省字符集，则 **mailx** 将尝试使用 **LANG** 的值查找用户语言环境的系统缺省字符集。如果该操作失败，则将使用 *us-ascii* 的缺省值。

cmd=*command* 设置 **pi**pe 命令的缺省命令。无缺省值。

conv=*conversion* 将 UUCP 地址转换为指定的地址样式。当前支持的唯一有效转换是 *Internet* ，它要求邮件传送程序符合用于电子邮件寻址的 RFC822 标准。缺省情况下禁用转换。另请参阅 **sendmail** 和 **-U** 命令行选项。

crt=*number* 将具有超过指定 *number* 的行的邮件，通过管道传递给由 **PAGER** 变量（缺省情况下为 **pg**）的值指定的命令（请参阅 *pg*(1) ）。缺省情况下禁用。

DEAD=*filename* 如果发生过早中断或传送错误，用于保存部分邮件的文件的名称。缺省值是 **$HOME/dead.letter** 。

debug 启用详细诊断信息供调试使用。不发送邮件。缺省值是 **nodebug** 。

dot 当处理来自终端的输入时，将输入行中的 ASCII 句点字符自身解释为文件结束标志。缺省值是 **nodot** 。

EDITOR=*command* 使用 **e**dit 或 **~e** 命令时要运行的命令。缺省值是 **ed** （请参阅 *ed*(1) ）。

encoding=*encoding* 设置当存在 8 位字符时要使用的缺省编码。允许值为 *quoted-printable* 、 *base64* 和 *8bit* 。对于 quoted-printable，也接受缩略语形式 *q-p* 。缺省值将根据 **charset** 的值进行确定。值

8bit 表示不进行编码。

escape=*c* 用 *c* 替换 ~ 转义字符。

folder=*directory* 用于保存标准邮件文件的目录。对于以加号 (+) 开头的用户指定文件名，可通过在该文件名之前加此目录名进行扩展，从而获取实际文件名。如果 *directory* 不以斜线 (/) 开头，则使用 **$HOME** 作为前缀。 **folder** 变量无缺省值。另请参阅下文的 **outfolder** 。

header 输入 **mailx** 时启用标题摘要的输出。缺省情况下启用。

hold 保留系统邮箱中已阅读的所有邮件，而不将它们放到标准 *mbox* 保存文件中。缺省值是 **nohold** 。

ignore 输入邮件时忽略中断。当通过嘈杂拨号线通信时很有用。缺省值是 **noignore** 。

ignoreeof 在邮件输入过程中忽略文件结束标志。必须在行中使用句点 (**.**) 自行终止输入，或者使用 **~.** 命令终止输入。缺省值是 **noignoreeof** 。另请参阅上文的 **dot** 。

keep 当 *mailbox* 为空时，将其截断为零长度，而不是将其删除。缺省情况下禁用。

keepsave 将已在其他文件中保存的邮件保留在系统邮箱中，而不是删除它们。缺省值是 **nokeepsave**。

MBOX=*filename* 用于保存已阅读邮件的文件的名称。 **x**it 命令会覆盖此功能，因为它同样可以将邮件明确地保存到另一个文件中。缺省值是 **$HOME/mbox** 。

metoo 通常，当对包含发件人的组（别名）进行扩展时，会将发件人从扩展中删除。设置此选项会使组中包含发件人。缺省值是 **nometoo** 。

mimeheader=*value* 发送邮件时添加或禁用 MIME 标题。 *value* 可以是 **yes** 或 **no** 。

LISTER=*command* 列出 **folder** 目录中的内容时要使用的命令及选项。缺省值是 *ls*(1) 。

NOMETAMAIL=*value*

要禁用 metamail 以读取 MIME 邮件，请将值设置为 **TRUE** 。缺省情况下不设置 **NOMETAMAIL** 变量。

onehop 当对最初发送至多个收件人的邮件作出响应时，通常强制其他收件人地址与原创者计算机相关，以便收到响应。此标志将禁止收件人地址发生变动，从而提高了网络效率，在该网络中，所有计算机可以直接向所有其他计算机发送邮件（即，无需中转）。

outfolder 将用于记录外发邮件的文件放置到 **folder** 变量指定的目录中。缺省值是 **nooutfolder** 。请参阅上文的 **folder** 以及 **S**ave 、 **C**opy 、 **fol**lowup 和 **F**ollowup 命令。

page 与 **pi**pe 命令一起使用，用于在通过管道发送的每个邮件之后插入一个换页符。缺省值是 **nopage** 。

PAGER=*command* 用作过滤器以便对输出进行分页的命令。它也可用于指定分页程序命令行选项（例如， **set PAGER="more -c"** ）。缺省值是 **pg** ，但许多用户偏爱 **more** （请参阅 *pg*(1) 和

more(1)）。

prompt=*string* 将命令模式提示符设置为 *string* 。缺省值是 **?** 。

quiet 输入 **mailx** 时，避免输出打开的邮件和版本。缺省值是 **noquiet** 。

record=*filename* Record all outgoing mail in *filename*. 缺省情况下禁用。另请参阅上文的 **outfolder** 。

replyto=*address* 指定要将响应发送至的地址。

save 发生中断或传送错误时，启用将邮件保存到 **dead.letter** 中。有关此文件的说明，请参阅 **DEAD** 。缺省情况下启用。

screen=*number* 为 **h**eaders 命令设置满屏标题的行数。

sendmail=*command* 用于发送邮件的备用命令。缺省值是 **mail** （请参阅 *mail*(1)）。

sendwait 在返回之前，等待后台邮件程序完成。缺省值是 **nosendwait** 。

SHELL=*command* 首选命令解释程序的名称。缺省值是用户的登录程序（请参阅 *passwd*(4)、 *shells*(4) 和 *chsh*(1)）。注释：在罕见情况下，用户的登录程序是脚本文件（而不是 Shell）， **mailx** 从该脚本文件中执行，然后 **mailx** 会要求用户明确地在 **$HOME/.mailrc** 文件中设置 **SHELL=/usr/bin/sh** 。

showto 当显示标题摘要，同时邮件是您发送的时侯，将输出收件人的名称，而不是作者的名称。

sign=*string* 当给定 **~a** （签名）命令时，邮件文本中要插入的变量。无缺省值（另请参阅波形符转义字符一节中的 **~i**）。

Sign=*string* 当给定 **~A** 命令时，邮件文本中要插入的变量。无缺省值（另请参阅波形符转义字符一节中的 **~i**）。

SMARTMAILER 当设置了 **SMARTMAILER** 时，各种命令都使用 **From:** 行，而不是缺省的 **From** 行。

toplines=*number* 要使用 **top** 命令输出的标题的行数。缺省值为 5。

VISUAL=*command* 首选屏幕编辑器的名称。缺省值是 **vi** （请参阅 *vi*(1)）。

下列环境变量取自执行环境，无法在 **mailx** 中进行更改。

HOME 用户主目录。通常是登录后随即进入的当前目录。

MAILRC 邮件程序启动文件的名称。缺省值是 **$HOME/.mailrc** 。

LC_COLLATE
LC_CTYPE 当调用命令解释程序（请参阅 **SHELL** 环境变量）时， **LC_COLLATE** 和 **LC_CTYPE** 会影响 **mailx** 。要确定 **LC_COLLATE** 和 **LC_CTYPE** 的行为，请参阅适用的命令解释程序的相应 Shell 联机帮助页。

LC_TIME **LC_TIME** 确定所显示的日期和时间字符串的格式及内容。如果在环境中未指定 **LC_TIME** ，或将其设置为空字符串，则 **LANG** 的值用作缺省值。如果未指定 **LANG**

或将其设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5) ）而不是 **LANG** 。如果任一国际化变量包含无效设置，则 **mailx** 就如所有国际化变量都设置为“C”那样工作。请参阅 *environ*(5) 。

TMPDIR 设置后， **TMPDIR** 环境变量会指定用于保存临时文件的目录，从而覆盖缺省目录 **/tmp**。

国际代码集支持

在邮件文本内，支持单字节字符代码集和多字节字符代码集。在标题中只能使用 7 位 USASCII 字符代码集（请参阅 *ascii*(5) ）中的字符。

警告

当 *command* 显示为有效时，并不总是允许使用参数。建议进行实验。

从执行环境中导入的内部变量不得是 **unset** 。

mailx 并不完全支持全部 Internet 寻址。新的国际化标准的落实尚需要一些时间。

标准邮件发送程序 *mail*(1) 将只由一个点 (**.**) 构成的行视为邮件结束。

使用两个单独的邮件程序同时访问同一邮件文件（通常是无意中通过两个单独的窗口），可能导致无法预见的结果。

包含多个词的参数必须用引号括起来。否则，可能会错误地解释它们。

文件

/var/mail/	邮局目录（模式 775，组 ID **mail** ）
/var/mail/*user*	*user* 的系统邮箱（模式 660， *user* 所有，组 ID **mail** ）
$HOME/.mailrc	个人启动文件
/usr/share/lib/mailx.rc	全局启动文件
$HOME/mbox	备用存储文件
/tmp/R[**emqsx**]*****	临时文件

另请参阅

chsh(1)、 echo(1)、 ed(1)、 lp(1)、 ls(1)、 mail(1)、 more(1)、 pg(1)、 sh(1)、 vi(1)、 passwd(4)、 shells(4)、ascii(5)、 environ(5)、 lang(5)

符合的标准

mailx：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

make - 维护、更新和重新生成程序组

概要

make [**-f** *makefile*] [**-bBdeiknpPqrsStuw**] [*macro_name***=***value*] [*names*]

说明

生成文件的结构

生成文件可以包含四种不同的行：目标行、 Shell 命令行、宏定义和包含行。

目标行

目标行包含由空白分隔的各目标的非空列表，后跟一个冒号 (**:**) 或双冒号 (**::**) ，然后是称为 相关项的必备文件的列表（可能为空）。将文件名作为相关项生成时，支持模式匹配表示法（请参阅 *regexp*(5) ）。

Shell 命令行

目标行上分号 (**;**) 后面的文本，以及以制表符开头的所有后续行，都是为更新目标而要执行的 Shell 命令（请参阅下面有关 **SHELL** 的环境小节）。不以制表符或 **#** 开头的第一行将开始一个新的目标定义、宏定义或包含行。通过使用 <反斜杠><换行符> 序列，可以使 Shell 命令跨行继续。

目标行及其关联命令行称为 规则。

宏

格式为 *string1* **=** *string2* 的行是宏定义。可以在生成文件中的任意位置定义宏，但是通常在开头将它们组合在一起。 *string1* 是宏名称； *string2* 是宏值。 *string2* 被定义为直到注释字符或未转义换行符的所有字符。紧邻 **=** 左侧和右侧的空格和制表符将被忽略。后来在生成文件中的任意位置（注释中除外）出现的 **$(***string1***)** 将被替换为 *string2* 。如果使用单字符宏名称，且没有替代序列，则圆括号是可选的。可以指定可选的替代序列 **$(***string1* [**:***subst1***=**[*subst2*]]**)** ，这导致在 *string1* 值的子字符串结尾出现的所有不重叠 *subst1* 都被替换为 *subst2* 。宏值中的子字符串由空白、制表符、换行符和行开头分隔。例如，如果

OBJS = file1.o file2.o file3.o

则

$(OBJS:.o=.c)

计算为

file1.c file2.c file3.c

宏值可以包含对其他宏的引用（请参阅警告一节）：

ONE =1

TWELVE = $(ONE)2

$(TWELVE) 的值设置为 **$(ONE)2** ，但是在目标行、命令行或包含行中使用它时，会将它扩展为 **12** 。如果 **ONE** 的值后来被生成文件靠下部分中或命令行上的其他定义更改，则对 **$(TWELVE)** 的任何引用都将反映该更改。

也可以在命令行上指定宏定义，这将覆盖生成文件中的任何定义。

（仅适用于 XPG4。命令行上的宏将添加到 **MAKEFLAGS** 环境变量中。在 **MAKEFLAGS** 环境变量中定义的宏（但是没有任何命令行宏）将宏添加到环境，并覆盖同名的任何现有环境变量）。

对于 **make** ，将自动定义某些宏（请参阅“内置宏”）。有关处理宏定义的顺序的讨论，请参阅环境小节。

条件宏定义可以覆盖分配给宏的值。条件宏定义采用以下格式： *target* **:=** *string1* **=** *string2* 。当处理与目标关联的目标行时，在条件宏定义中指定的宏值有效。如果以前定义了 *string1* ，则 *string1* 的新值将覆盖以前的定义。在处理目标或其任何相关项时， *string1* 有效。

包含行

如果字符串 **include** 在生成文件中显示为某行的前七个字母，并且后跟一个或多个空格或制表符，则假定该行的其余部分是文件名，并在扩展文件名中的任何宏之后， **make** 的当前调用将该部分作为其他生成文件进行读取和处理。

如果目标没有与其关联的显式命令且定义了 **.DEFAULT** 目标，则 **make** 的缺省行为是使用 **.DEFAULT** 内置目标。请参阅内置目标小节。

常规说明

make 执行以前在生成文件中放置的命令以更新一个或多个目标名称。目标名称通常为程序名称。如果未指定 **-f** 选项，则尝试文件名 **makefile** 、 **Makefile** 、 **s.makefile** 、 **SCCS/s.makefile** 、 **s.Makefile** 和 **SCCS/s.Makefile** （按照该顺序）。如果指定了 **-f -** ，则使用标准输入。可以指定多个 **-f** 选项。生成文件的参数是按指定顺序处理的。 **-f** 和文件名之间 **must** 有一个空格，而且多个生成文件名称必须均在其前面有自己的 **-f** 选项。生成文件的内容将覆盖内置规则和宏（如果它们存在）。

如果未在命令行上指定目标名称，则 **make** 更新（第一个）生成文件中不是推断规则的的第一个目标。在下列两种情况下将更新目标：第一种情况，它依赖于比目标新的文件；第二种情况，它依赖于与目标具有相同修改时间的文件。缺少的文件被认为已过期。在更新目标之前，如果需要，则以递归方式更新目标的所有相关项。这将影响目标的相关性树的深度第一型更新。

如果目标没有在目标行上分隔符之后指定任何相关项（ 显式相关项），则在目标过期时，将执行与该目标关联的任何 Shell 命令。

目标行上的一个或多个目标名称和任何显式相关项名称之间，可以是单冒号或双冒号。目标名称可以出现在多个目标行上，但是所有那些行都必须属于相同的（单冒号或双冒号）类型。对于通常的单冒号情况，至多其中一个目标行可以具有与它关联的显式命令。如果目标已过期，且在任一行上具有其任一相关项，则将执行显式命令（如果指定了它们），否则可能执行缺省规则。对于双冒号情况，显式命令可以与包含目标名称的多个目标行关联；如果目标已过期，且在特定行上具有任一相关项，则将执行该行的命令。也可能执行内置规则。

目标行及其关联的 Shell 命令行也称为 规则。散列标记 **(#)** 和换行符围绕生成文件中任意位置（规则中除外）的注释。规则中的注释取决于 **SHELL** 宏的设置。

以下生成文件指出 **pgm** 取决于两个文件： **a.o** 和 **b.o** ，而这两个文件又取决于其对应源文件（ a.c 和 **b.c** ）和通用文件 **incl.h** ：

OBJS = a.o b.o

```
pgm: $(OBJS)
    cc $(OBJS) -o pgm

a.o: incl.h a.c
    cc -c a.c

b.o: incl.h b.c
    cc -c b.c
```

命令行由其自己的 Shell 每次执行一个。每个命令行可以具有下列一个或多个前缀： **-** 、 **@** 或 **+** 。这些前缀在下面进行说明。

返回非零状态的命令通常终止 **make** 。 **-i** 选项或生成文件中特殊目标 **.IGNORE** 的存在可导致 **make** 继续执行生成文件，而不管有多少命令行导致了错误（尽管仍然在标准输出上输出错误消息）。如果在命令行的开头存在 **-**，则该行返回的任何错误都将输出到标准输出，但 **make** 不终止。可以使用前缀 **-** 有选择地忽略生成文件中的错误。如果指定了 **-k** 选项，且命令行返回错误状态，则放弃对当前目标执行操作，但是继续对不依赖该目标的其他分支执行操作。如果在 **MAKEFLAGS** 环境变量中存在 **-k** 选项，则通过指定 **-S** 选项可以将处理返回到缺省状态。

-n 选项指定输出命令行但不执行它。但是，如果命令行包含字符串 **$(MAKE)** 或 **${MAKE}** 或者将 **+** 作为前缀，则将始终执行该行（请参阅环境下对 **MAKEFLAGS** 宏的讨论）。 **-t** （处理）选项更新文件的修改日期，但不执行任何命令。

通常在执行命令行之前将其输出，但是，如果命令行开头有 **@** ，则将禁止输出。 **-s** 选项或生成文件中存在的特殊目标 **.SILENT:** 禁止输出所有命令行。可以使用 **@** 有选择地关闭输出。由 **make** 输出的所有内容（初始制表符除外）将直接传递到 Shell，而不会进行更改。因此，

```
echo a\
b
```

产生

```
ab
```

就像 Shell 那样。

-b 选项允许运行旧的生成文件（为旧版 **make** 编写的生成文件）且不导致错误。旧版本的 **make** 假定，如果目标没有与其关联的任何显式命令，则用户希望命令为空，并且将不执行可能已定义的任何 **.DEFAULT** 规则。缺省情况下，当前版本的 **make** 在该模式下运行。但是，当前版本的 **make** 提供了 **-B** 选项（它将该模式关闭），以便在目标没有与其关联的显式命令且定义了 **.DEFAULT** 规则的情况下，执行 **.DEFAULT** 规则。请注意， **-b** 和 **-B** 选项对目标的适当推断规则的搜索以及可能查找和执行不起作用。始终执行对除 **.DEFAULT** 之外的内置推断规则的搜索。

除非目标依赖于特殊名称 **.PRECIOUS** ，否则信号 **SIGINT** 、 **SIGQUIT** 、 **SIGHUP** 和 **SIGTERM** （请参阅 *signal*(5) ）会导致删除该目标。

选项

以下是所有选项和一些特殊名称的简短说明。选项可以按任何顺序出现。可以分别指定它们，也可以与一个 **-** （**-f** 选项除外）一起指定。

-b 旧版本 (v7) 生成文件的兼容模式。缺省情况下打开该选项。

-B 关闭旧版本 (v7) 生成文件的兼容模式。

-d 调试模式。输出有关文件和检查时间的详细信息。（这是非常详细的，用于调试 **make** 命令本身）。

-e 环境变量覆盖生成文件内的分配。

-f *makefile* 说明文件名，称为生成文件。文件名 **-** 表示标准输入。生成文件的内容将覆盖内置规则和宏（如果它们存在）。请注意，在 **-f** 和 *makefile* 之间 **must** 有一个空格。允许使用该选项的多个实例（**-f -** 除外），将按指定的顺序处理它们。

-i 忽略由所调用命令返回的错误代码。如果特殊目标名称 **.IGNORE** 出现在生成文件中，则也将进入该模式。

-k 当命令返回非零状态时，将放弃对当前条目执行操作，但继续对不依赖该目标的其他分支执行操作。这与 **-S** 相反。如果同时指定了 **-k** 和 **-S** ，则使用指定的最后一个选项。

-n 无执行模式。输出命令，但不执行它们。输出以 **@** 开头的偶数行。但是，将执行包含字符串 **$(MAKE)** 或 **${MAKE}** 的行或者以 **+** 为命令前缀的行。

-p 将宏定义和目标说明的完全集写入标准输出。

-P 一次并行更新多个目标。并发更新的目标数由环境变量 **PARALLEL** 和在生成文件中存在的 **.MUTEX** 指令确定。

-q 问题。 **make** 命令返回零或非零状态代码，具体取决于目标文件是否为最新。不使用该选项更新目标。

-r 清除后缀列表且不使用内置规则。

-s 静默模式。在执行命令行之前，不将它们输出到标准输出。如果特殊目标名称 **.SILENT** 出现在生成文件中，则也将进入该模式。

-S 如果执行命令使目标为最新时出现错误，则终止。这是缺省值，它与 **-k** 相反。如果同时指定了 **-k** 和 **-S** ，则使用指定的最后一个选项。这样就可以覆盖 **MAKEFLAGS** 环境变量中存在的 **k** 标志。

-t 处理目标文件（使它们为最新）而不是发出通常的命令。

-u 无条件地 **make** 目标，忽略所有时间戳。

-w 禁止警告消息。将不影响致命消息。

[*macro_name*=*value*]
可以指定零个或更多命令行宏定义。请参阅宏小节。

[*names*] 在生成文件中出现的零个或更多目标名称。这样指定的每个目标由 **make** 更新。如果未指定名称，则 **make** 更新生成文件中不是推断规则的第一个目标。

并行建立

如果 **make** 是使用 **-P** 选项调用的，则它将尝试以并行方式一次构建多个目标。（这是通过使用允许多个进程同时运行的标准 UNIX 系统进程机制实现的）。对于在上一小节的示例中显示的生成文件，它将创建进程以并行构建 **a.o** 和 **b.o** 。在这些进程完成后，它将构建 **pgm** 。

make 将尝试并行构建的目标数由环境变量 **PARALLEL** 的值确定。如果调用 **-P** ，但是未设置 **PARALLEL**，则 **make** 将尝试并行构建不超过两个的目标。

您可以使用 **.MUTEX** 指令将一些指定目标的更新串行化。当两个或更多目标修改公共输出文件（如将模块插入到归档或创建同名的中间文件，如 **lex** 和 **yacc** 完成的那样）时，这是很有用的。如果上一小节中的生成文件包含以下形式的 **.MUTEX** 指令

.MUTEX: a.o b.o

则它将阻止 make 并行构建 a.o 和 b.o。

环境

在环境中定义的所有变量（请参阅 *environ*(5)）都由 **make** 读取，并被视为宏定义进行处理，但始终被忽略的 **SHELL** 环境变量除外。 **SHELL** 环境变量的值将不用作宏，且不通过在生成文件中或命令行上定义 **SHELL** 宏来进行修改。无定义的变量或字符串定义为空的变量由 **make** 包括在内。

存在四个可能的宏定义源，按以下顺序读取：内部（缺省值）、当前环境、生成文件和命令行。由于采用该处理顺序，生成文件中的宏分配将覆盖环境变量。 **-e** 选项允许环境覆盖生成文件中的宏分配。命令行宏定义始终覆盖任何其他定义。

MAKEFLAGS 环境变量由 **make** 处理的前提是，它包含为命令行定义的任何合法输入选项（ **-f** 、 **-p** 和 **-d** 除外）。也可以在生成文件中指定 **MAKEFLAGS** 变量。

（仅适用于 XPG4。生成文件中的 **MAKEFLAGS** 将替换 **MAKEFLAGS** 环境变量。命令行选项的优先级高于 **MAKEFLAGS** 环境变量）。

如果在这两个位置中均未定义 **MAKEFLAGS** ，则 **make** 将为自己构建变量，将在命令行上指定的选项和任何缺省选项放置在该变量中，并将该变量传递到命令的调用。因此， **MAKEFLAGS** 始终包含当前的输入选项。这证明对于递归的 **make** 是很有用的。甚至在指定 **-n** 选项时，也将执行包含字符串 **$(MAKE)** 或 **${MAKE}** 的命令行；因此，用户可以按递归方式在整个软件系统上执行 **make -n** ，以查看已经执行的内容。这可能是因为 **-n** 已放入 **MAKEFLAGS** 并传递到 **$(MAKE)** 或 **${MAKE}** 的递归调用。这是为软件项目调试所有生成文件而不实际执行任何命令的一种方法。

规则中的每个命令都将提供给要执行的 Shell。所用的 Shell 是 Shell 命令解释程序（请参阅 *sh*(1)）或 **SHELL** 宏在生成文件中指定的命令解释程序。为确保每次执行生成文件时使用同一 Shell，以下行:

SHELL=/usr/bin/sh

或合适的等效项应该放置在生成文件的宏定义部分中。

后缀

目标和（或）相关项的名称通常带有后缀。有关某些后缀的知识已内置在 **make** 中，用于识别为更新目标而要应用的相应推断规则（请参阅推断规则小节）。缺省的后缀列表当前如下：

.o .c .c˜ .B .B˜ .cxx .cxx˜ .cpp .cpp˜ .cc .cc˜
.y .y˜ .l .l˜ .L .L˜ .Y .Y˜ .s .s˜ .sh .sh˜
.h .h˜ .H .H˜ .p .p˜ .f .f˜ .r .r˜

这些后缀定义为特殊内置目标 **.SUFFIXES** 的相关项。这是由 **make** 自动完成的。

在生成文件中可以将其他后缀指定为 **.SUFFIXES** 的相关项列表。这些其他后缀的值将被添加为缺省值。多个后缀列表可累积。后缀列表的顺序是很重要的（请参阅推断规则小节）。如果用户希望更改后缀的顺序，则他必须首先用空的相关项列表定义 **.SUFFIXES** （这样可清除 **.SUFFIXES** 的当前值），然后按所需顺序用这些后缀定义 **.SUFFIXES** 。可以通过以下命令行在任何计算机上显示在 **make** 中内置的后缀的列表：

make -fp - 2>/dev/null </dev/null

可以通过以下命令行显示并入名为 **mymakefile** 的给定生成文件中定义的内置后缀的列表：

make -fp mymakefile 2>/dev/null </dev/null

推断规则

某些目标或相关项名称（如以 **.o** 结尾的名称）具有可推断相关项，如 **.c** 和 **.s** 等。如果生成文件中未出现此类名称的更新命令，并且存在可推断相关项文件，则将编译该相关项文件以更新目标。在这种情况下， **make** 的推断规则允许从其他文件构建文件，方法是检查后缀并确定要使用的相应推断规则。当前存在为以下项定义的缺省推断规则：

单后缀规则

.c .c˜
.B .B˜ .cxx .cxx˜ .cpp .cpp˜ .cc .cc˜
.sh .sh˜
.p .p˜
.f .f˜
.r .r˜

双后缀规则

.c.o .c˜.o .c˜.c .c.a .c˜.a
.B.o .B˜.o .B˜.B .B.a .B˜.a
.cxx.o .cxx˜.o .cxx˜.cxx .cxx.a .cxx˜.a
.cpp.o .cpp˜.o .cpp˜.cpp .cpp.a .cpp˜.a
.cc.o .cc˜.o .cc˜.cc .cc.a .cc˜.a

.s.o .s˜.o .s˜.a
.p.o .p˜.o .p˜.p .p.a .p˜.a
.f.o .f˜.o .f˜.f .f.a .f˜.a
.r.o .r˜.o .r˜.r .r.a .r˜.a
.y.o .y˜.o .y.c .y˜.c
.l.o .l˜.o .l.c
.h˜.h .H˜.H .hxx˜.hxx .hpp˜.hpp
.B.o .B˜.o .B.a .B˜.a
.L.B .L.o .L˜.B .L˜.L .L˜.o
.Y.B .Y.o .Y˜.B .Y˜.Y .Y˜.o

双后缀推断规则 (**.c.o**) 定义如何从 **x.c** 构建 **x.o** 。单后缀推断规则 (**.c**) 定义如何从 **x.c** 构建 **x** 。实际上，第一个后缀为空。对于仅从一个源文件（例如，Shell 过程和简单 C 程序）构建目标，单后缀规则是很有用的。

上述规则中的波形符是指 SCCS 文件（请参阅 *sccsfile*(4) ）。因此，规则 **.c˜.o** 会将 SCCS C 源文件转换为对象文件 (**.o**) 。由于 SCCS 文件的 **s.** 是前缀，因此它与 **make** 的后缀观点不兼容。因此，波形符是将任何文件引用更改为 SCCS 文件引用的一种方法。

从后缀为 **.c** 的文件创建后缀为 **.o** 的文件的规则被指定为具有 **.c.o** （作为目标）且没有相关项的条目。与目标关联的 Shell 命令定义从 **.c** 文件生成 **.o** 文件的规则。将不包含斜线且以点开头的任何目标名称识别为推断（隐式）规则而不是目标（显式）规则。具有一个点的目标是单后缀推断规则；具有两个点的目标是双后缀推断规则。用户可以在生成文件中定义其他推断规则，以及重新定义或取消缺省的推断规则。

用于将 **.c** 文件更改为 **.o** 文件的缺省推断规则可能与如下所示类似：

```
.c.o:
  $(CC) $(CFLAGS) -c $<
```

用于将 **yacc** 文件更改为 C 对象文件的缺省推断规则可能与如下所示类似：

```
.y.o:
  $(YACC) $(YFLAGS) $<
  $(CC) $(CFLAGS) -c y.tab.c
  rm y.tab.c
  mv y.tab.o $@
```

某些宏在缺省推断规则中使用，以允许在任何得到的命令中包括可选内容。例如， **CFLAGS** 、 **LDFLAGS** 和 **YFLAGS** 分别用于 *cc*(1) 、 *lex*(1) 和 *yacc*(1) 的编译程序选项。 **LDFLAGS** 通常用于指定链接程序（或加载程序）选项。这些宏是由 **make** 自动定义的，但是用户可以在生成文件中重新定义它们。

依照惯例，宏 **LIBS** 用于指定在编译的链接阶段包括任何特殊库的顺序。要为一组特定的库指定特定的包括顺序，则 **.c** 文件的现有单后缀规则

```
$(CC) $(CFLAGS) $< $(LDFLAGS) -o $@
```

可以重新定义为

```
$(CC) $(CFLAGS) $< $(LDFLAGS) -o $@ $(LIBS)
```

以及在生成文件中定义 **LIBS** 。

还存在一些在推断规则中使用的特殊内置宏（ @、 < ）。请参阅内置宏小节。

如果目标没有显式相关项，或者如果相关项也没有使其与关联显式规则匹配的目标，则 **make** 将查找同时与目标（相关项）的后缀（它可能为空）和与规则的其他后缀匹配的文件匹配的第一个推断规则。由于它通过遍历 **.SUFFIXES** 值的列表从前向后进行该搜索，因此定义 **.SUFFIXES** 的顺序是很重要的。

要在任何计算机上输出已编译到 **make** 中的规则，请键入：

```
make -fp - 2>/dev/null </dev/null
```

由于 **make** 定义推断规则 **.c.o** ，因此可以更简单地重新编写常规说明小节中的示例：

```
OBJS = a.o b.o
  pgm: $(OBJS)
    cc $(OBJS) -o pgm
  $(OBJS): incl.h
```

库

如果目标或相关项的名称包含圆括号，则假定它是归档库，圆括号内的字符串指库内的成员。因此， **lib(file.o)** 和 **$(LIB)(file.o)** 都是指包含 **file.o** 的归档库（这假定以前已经定义了 **LIB** 宏）。表达式 **$(LIB)(file1.o file2.o)** 无效。与归档库有关的规则的格式为 **.*xx*.a** ，其中 *xx* 是要从其建立归档成员的后缀。当前实现的令人遗憾的副作用要求 *xx* 与归档成员的后缀不同。因此，不能让 **lib(file.o)** 显式依赖于 **file.o** 。归档接口的最常见用法如下所示。在此处，我们假定源文件都是 C 类型源：

```
lib: lib(file1.o) lib(file2.o) lib(file3.o)
    @echo lib is now up-to-date
.c.a:
    $(CC) -c $(CFLAGS) $<
    ar rv $@ $*.o
    rm -f $*.o
```

（有关 **<** 、 **@** 和 ***** 符号的说明，请参阅内置宏小节）。实际上，上面列出的 **.c.a** 规则已内置到 **make** 中，在该示例中是不需要的。该规则接着又依次应用于 **lib** 的每个相关项。以下示例更有效地实现这一点：

```
lib:  lib(file1.o) lib(file2.o) lib(file3.o)
    $(CC) -c $(CFLAGS) $(?:.o=.c)
    ar rv lib $?
    rm $?
    @echo lib is now up-to-date
.c.a:;
```

在此处使用了宏中的替换。 **$?** 列表定义为其 C 源文件已过期的一组对象文件名（在 **lib** 内）。替换序列将 **.o** 转

换为 **.c** （遗憾的是，到目前为止无法转换为 **.c~** ；但是，在将来这也许是可能的）。另请注意，如果禁用 **.c.a** 规则，则将逐个创建并归档每个对象文件。该特定结构大大加快了归档库的维护速度，但是，如果归档库同时包含汇编程序和 C 程序，则会变得非常麻烦。

包含符号定义的归档成员是通过符号名 **lib((***entry_name***))** 前后的双圆括号指定的，否则将如上所述进行处理。

内置目标

make 具有有关一些特殊目标的知识。这些特殊目标必须在生成文件中指定才能生效（但 **.SUFFIXES** 除外，它是由 **make** 自动设置的，但是用户可以更改它）。

.DEFAULT 如果必须创建某个文件但是不存在它的显式命令或相关内置规则，则在生成文件中定义了 **.DEFAULT** 的情况下，使用与目标名称 **.DEFAULT** 关联的命令。 **.DEFAULT** 没有任何显式相关项。

.PRECIOUS 在收到 **QUIT** 、 **INTERRUPT** 、 **TERMINATE** 或 **HANGUP** 时，不删除该目标的相关项。

.SILENT 与 **-s** 选项的作用相同。不需要指定相关项或显式命令。

.IGNORE 与 **-i** 选项的作用相同。不需要指定相关项或显式命令。

.SUFFIXES 将 **.SUFFIXES** 的显式相关项添加到已知后缀的内置列表，并与推断规则联合使用。如果 **.SUFFIXES** 没有任何相关项，则清除已知后缀的列表。没有与 **.SUFFIXES** 关联的命令。

.MUTEX 串行化指定目标的更新（请参阅并行建立一节）。

内置宏

有五个在内部维护的宏，它们对于编写构建目标的规则是很有用的。为了清晰定义这些宏的含义，需要对术语 目标和 相关项进行一些说明。当 **make** 更新目标时，它可能实际生成一系列要更新的目标。在将任何规则（显式或隐式）应用于目标以更新它之前，在目标的每个相关项上都将发生递归。相关项在递归时将变成目标本身，而且可能具有或生成它自己的相关项，接着又依次进行递归，直到找到没有相关项的目标，此时递归停止。并不是由 **make** 处理的所有目标都在生成文件中显示为显式目标；其中一些目标是生成文件中的显式相关项，而其他目标是在 **make** 以递归方式更新目标时生成的隐式相关项。例如，在执行以下生成文件时：

```
pgm:  a.o b.o
  cc a.o b.o -o pgm
```

生成要建立的以下目标系列：

--- pgm 带有两个相关项和一个要遵循的显式规则

--- a.o （以递归方式）带有与隐式规则 **.c.o** 匹配的 **a.c** 的隐式相关项

--- a.c （以递归方式）没有隐式相关项且没有隐式规则。这将停止递归，而且仅返回文件 **a.c.** 的上次修改时间

--- b.o （以递归方式）带有与隐式规则 **.c.o** 匹配的 **b.c** 的隐式相关项

--- b.c （以递归方式）没有隐式相关项且没有隐式规则。这将停止递归，而且仅返回文件 **b.c** 的上次修改时间。

在下面的定义中， *target* 一词是指在生成文件中指定的目标、在生成文件中指定的显式相关项（当 **make** 在其上递归时，它将变成目标）或隐式相关项（它作为查找推断规则和与目标的后缀匹配的文件的结果生成；当 **make** 在其上递归时，它将变成目标）。“相关项”一词是指在生成文件中为特定目标指定的显式相关项，或者作为查找相应推断规则和与目标后缀匹配的对应文件而生成的隐式相关项。

将目标规则看作用户为特定目标名称指定的规则，并将推断规则看作用户或 **make** 为特定类别的目标名称指定的规则，可能是有帮助的。牢记当 **make** 同时在显式相关项和隐式相关项上递归时目标名称及其对应相关项名称会发生改变，以及推断规则仅应用于隐式相关项或没有在生成文件中为其定义目标规则的显式相关项，也可能是有帮助的。

$@** **$@ 宏是当前目标的完整目标名称，或者是库归档目标的归档文件名部分。对于目标规则和推断规则，均计算该宏。

$% 仅当当前目标是格式为 **libname(***member***.o)** 或 **libname((***entry***))** 的归档库成员时，才计算 **$%** 宏。在这些情况下， **$@** 的计算结果为 **libname** ， **$%** 的计算结果为 **member.o** 或包含符号 **entry** 的对象文件。对于目标规则和推断规则，均计算 **$%** 。

$?** **$? 宏是相对当前目标已过期的相关项的列表；从本质上说，就是已经重建的那些模块。对于目标规则和推断规则，均计算该宏，但是通常仅在目标规则中使用它。 **$?** 在推断规则中仅计算为一个名称，但是在目标规则中可能计算为多个名称。

$<** 在推断规则中， **$< 计算为与隐式规则（与所建立的目标的后缀匹配）对应的源文件名。换句话说，它是相对于目标已过期的文件。在 **.DEFAULT** 规则中， **$<** 宏计算为当前目标名称。仅对推断规则计算 **$<** 。因此，在 **.c.o** 规则中， **$<** 宏将计算为 **.c** 文件。从 **.c** 文件建立优化的 **.o** 文件的示例如下：

```
.c.o:
    cc -c -O $*.c
```

或：

```
.c.o:
    cc -c -O $<
```

$*** 宏 **$* 是已删除后缀的当前目标名称。仅为推断规则计算它。

这五个宏可以具有其他形式。将大写的 **D** 或 **F** 追加到这五个宏中的任何一个时，含义将更改为“目录部分”（对于 **D** ）和“文件部分”（对于 **F** ）。因此， **$(@D)** 是指字符串 **$@** 的目录部分。如果没有目录部分，则将生成 **./** 。当 **$?** 宏包含多个相关项名称时， **$(?D)** 将扩展为目录名称部分的列表， **$(?F)** 将扩展为文件名部分的列表。

除了上面列出的内置宏外，其他常用的宏由 **make** 定义。这些宏在缺省推断规则中使用，并且可以通过 **-p** 选项进

行显示。这些宏可以在生成文件的目标规则中使用。也可以在生成文件中重新定义它们。

$$@** **$$@** 宏 仅在相关性行上才有意义。该形式的宏称为 “动态相关项”，因为它们是在实际处理相关项时计算的。 **$$@** 计算为与 **$@does**on**a**command**line 完全相同；即当前的目标名称。对于构建大量可执行文件（其中每个文件只有一个源文件），该宏是很有用的。例如，使用同一规则可以构建下列所有 HP-UX 命令：

```
CMDS = cat echo cmp chown
$(CMDS) : $$@.c
   $(CC) -O $? -o $@
```

如果该生成文件是使用 **make cat echo cmp chown** 调用的，则 **make** 将使用常规规则依次构建每个目标， **$$@** 在 **cat** 是目标时计算为 **cat** ，在目标为 **echo** 时计算为 **echo** ，依此类推。

动态相关宏也可以采用 F 形式 **$$(@F)** （它表示 **$$@** 的文件名部分）。如果目标包含路径名，则这是很有用的。例如：

```
INCDIR = /usr/include
INCLUDES = $(INCDIR)/stdio.h \
      $(INCDIR)/pwd.h   \
      $(INCDIR)/dir.h   \
      $(INCDIR)/a.out.h
$(INCLUDES) : $$(@F)
   cp $? $@
   chmod 0444 $@
```

特殊宏

VPATH 宏允许 make 在冒号分隔的目录列表中搜索相关项。形式为 **VPATH=** *path1***:***path2* ... 的行会导致 make 首先在当前目录中搜索相关项，如果找不到相关项，则 make 搜索 *path1* 并继续下去，直到搜索完在 VPATH 宏中指定的目录。

外部语言环境影响

环境变量

LANG 为没有设置或为空的国际化变量提供了缺省值。如果未设置 **LANG** 或为空，则使用缺省值“C”（请参阅 *lang*(5) ）。如果任一国际化变量中包含无效设置，则 **make** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

LC_ALL 如果设为非空字符串值，则会覆盖其他所有国际化变量的值。

LC_CTYPE 用于确定将文本解释为单字节字符和（或）多字节字符，将字符的类别解释为可输出，以及确定与正则表达式中的字符类表达式相匹配的字符。

LC_MESSAGES 确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息以及写入到标准输出的信息消息的格式和内容。

NLSPATH 用于确定消息清单的位置，以便处理 **LC_MESSAGES**。

PROJECTDIR 提供用于搜索在当前目录中未找到的 SCCS 文件的目录。在下面的所有情况下，将在标识目录的目录 **SCCS** 中搜索 SCCS 文件。如果 **PROJECTDIR** 的值以斜线开头，则认为它是绝对路径名；否则，将检查该名称的用户的主目录，以查找子目录 **src** 或 **source** 。如果找到这样的目录，则使用它。否则，该值将用作相对路径名。

如果未设置 **PROJECTDIR** 或它具有空值，则将首先搜索当前目录，然后在当前目录的 **SCCS** 目录中进行搜索。**PROJECTDIR** 的设置影响在具有名为 **SCCS** 的组件的文件的该实用程序说明的其余部分中列出的所有文件。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

返回值

如果成功完成，则 **make** 返回 0；如果出现错误，则返回大于 0 的值。如果指定了 **-q** 选项，则 **make** 在目标是最新时返回 0；在目标不是最新时返回大于 0 的值。

举例

如果 program.c 存在于当前目录中，则下例在没有生成文件的情况下从 C 源代码文件创建可执行文件：

make program

下例显示指定的多个生成文件和已定义的一些命令行宏，并更新模块 1 中的第一个目标：

make -f module1 -f module2 RELEASE=1.0 CFLAGS=-g

下例更新当前驻留在当前目录中的缺省生成文件中的两个目标：

make clobber prog

下例更新指定生成文件中的 prog 目标，允许环境变量覆盖生成文件中的任何公共变量，清除内置后缀列表并忽略内置规则，以及输出详尽的调试信息：

make -erd -f module1 prog

警告

请留意其访问时间、修改时间和上次更改时间无法由 **make** 进程更改的任何文件（如包含文件）。例如，如果某个程序依赖于包含文件（它又依赖于其他包含文件），并且这两个文件之一或两者已过期，则 **make** 将尝试在每次运行时都更新这些文件，这样就不必 重新生成依赖于包含文件的最新文件。解决方法是在运行 **make** 之前，使用 **touch** 命令手动更新这些文件（请参阅 *touch*(1)）。

一些命令会不适当地返回非零状态；使用 **-i** 可以克服该困难。

包含字符 **=**、 **:**、 **@** 和 **$** 的文件名不起作用。

由 Shell 直接执行的内置命令（如 **cd** ，请参阅 *cd*(1)）在跨 **make** 中的换行符时不起作用。

语法 **(lib(file1.o file2.o file3.o)** 是非法的。

您不能从 **file.o** 构建 **lib(file.o)** 。

宏 **$(a:.o=.c˜)** 不起作用。

扩展的目标行包含的字符不能超过 16384 个（包括结尾换行符）。

如果当前目录中不存在生成文件，则键入

make filename

会导致 **make** 尝试从 **filename.c** 构建 **filename**

如果在 Shell 脚本中使用计算为 NULL 的括在引号中的参数（如 **$@** ）调用 **make** ，则 **make** 将失败。

相关内容

NFS 警告

在比较位于不同 NFS 服务器上的文件的修改时间时，如果服务器上的时钟未进行同步，则 **make** 的行为是不可预知的。

文件

[Mm]akefile
s.[Mm]akefile
SCCS/s.[Mm]akefile

另请参阅

cc_bundled(1)、 cd(1)、 lex(1)、 mkmf(1)、 sh(1)、 environ(5)、 lang(5)、 regexp(5)。

《A Nutshell Handbook, Managing Projects With Make》， Steve Talbot，第 2 版，O’Reilly & Associates, Inc.， 1986 年。

符合的标准

make： SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

makekey - 生成加密密钥

概要

/usr/lbin/makekey

说明

makekey 通过增加搜索密钥空间所需的时间，来提高使用密钥的加密方案的效用。该命令从其标准输入中读取 10 个字节，然后向其标准输出中写入 13 个字节。输出与输入之间的对应关系应尽可能难以计算（即，需要用 1 秒内很精确的小数）。

前八个输入字节（ 输入关键字）可以是任意的 ASCII 字符。后两个字节 (*salt*) 是从数字、 **.** 、 **/** 和大小写字母字符集中精心选择出来的。 Salt 字符还作为输出的前两个字符。其余的 11 个输出字符从 salt 所在的字符集中进行选择，构成 输出关键字。

执行的转换的实质是：使用 Salt 从 4,096 台加密机中选择一个，这些加密机全部基于 National Bureau of Standards 规定的 DES 算法，但其解密方法多达 4,096 种。使用 输入关键字作为密钥时，会将一个常量字符串输入到加密机中，并循环多次。生成的 64 位被分配到结果中的 66 位 输出关键字中。

makekey 用于执行加密的程序（例如， *ed*(1) 和 *crypt*(1) ）。通常，此命令的输入和输出将是管道。

另请参阅

crypt(1)、ed(1)、passwd(4)。

名称

man - 按关键字查找手册信息；输出手册条目

概要

man [**-M** path] **-k** *keyword*...

man [**-M** path] **-f** *file*...

man [**-**] [**-M** path] [**-T** macro-package] [*section* [*subsection*]] *entry_name*...

说明

man 从 HP-UX 手册页访问信息。该命令可用于:

- 列出其单行说明中包含任何指定的一组关键字的所有手册条目。
- 显示或输出名称所指定的条目的单行说明。
- 按条目名称搜索联机手册目录，并显示或输出指定的一个或多个条目。
- 搜索指定的联机手册节（目录），并显示或输出该节中指定的一个或多个条目。

按关键字搜索条目名称（第一种形式）

上述第一种形式在各条目的单行说明中搜索指定的关键字。参数如下:

-k *keyword* 当 **-k** 后跟一个或多个关键字时， **man** 将输出每个手册条目的单行说明，该单行说明中包含与指定的一个或多个关键字相匹配的文本（与 *grep*(1) 的行为类似）。关键字由空白（空格或制表符）分隔。

只有当存在文件 **/usr/share/lib/whatis** 时，才能使用此选项。可以通过运行 *catman*(1M) 来创建 **/usr/share/lib/whatis** 。

获得条目的单行说明（第二种形式）

上述第二种形式可查找指定单独条目的单行说明，并显示或输出这些说明。参数如下:

-f *file* 当 **-f** 后跟一个或多个文件名时， **man** 将输出找到的其名称与 *file* 匹配的每个手册条目的单行说明。在指定两个或多个文件时， *file* 参数由空白（空格或制表符）分隔。如果与 *file* 匹配的条目名称存在于两节或多节中，则将输出每个匹配文件名的单行说明。

只有当存在文件 **/usr/share/lib/whatis** 时，才能使用此选项。可以通过运行 *catman*(1M) 来创建 **/usr/share/lib/whatis** 。

查看单独的手册条目（第三种形式）

上面所示的第三种形式用于查看一个或多个单独的手册条目。此形式的 **man** 可识别下列参数:

- （可选）当存在 **-** 参数时， **man** 将已设置格式的手册条目直接发送到标准输出，而不通过 **PAGER** 环境变量指定的输出过滤器处理它。

-M *path* 更改手册页的搜索路径。 *path* 是包含手册页目录子树的目录的冒号分隔列表。当与 **-k** 或 **-f** 选项一起使用时， **-M** 选项必须位于其他两个选项之前。

-T *macro-package*

man 使用宏程序包而不是在 **/usr/share/lib/tmac/tmac.an** 中定义的标准 -man 宏来设置手册页格式。

将 **-T** 选项指定为 **man** 时，必须提供完整路径。例如：

man -T /usr/share/lib/tmac/tmac.s ls

section[*subsection*]

（可选）在指定节中搜索给定的 *entry_name* 。 *section* 指定单个节编号或 **local** 、 **new** 、 **old** 或 **public** 之一，来搜索指定的一个或多个条目。 *section* 对应于《HP-UX 参考手册》中出现相应条目的节编号。它可以后跟可选的大写/小写子节标识符（如 **3C** ，指示第 3 节中的库例行程序）。 **3** 、 **3c** 和 **3C** 被解释为等效，因为第 3 节中的所有手册条目都存储在同一目录或相关目录（如 **/usr/share/man/man3.Z** 和 **/usr/share/man/man3** ）中。但是，如果条目位于第 1M 节中，则必须将 *section* 指定为 **1m** 或 **1M** 。

entry_name

搜索特定条目名称，其中 *entry_name* 是手册条目的名称，不包含其节编号后缀。除了超过 11 个字符的名称外， *entry_name* 与在每页顶部列出的手册条目名称完全相同，或者与相应手册条目的单行说明中左侧部分的关键字之一相同。

如果 *entry_name* 的长度超过 11 个字符，则 **man** 将首先搜索完整长度的 *entry_name* 。如果未找到这种名称，则会将 *entry_name* 截断至 11 个字符，以确保为 14 字符源文件名中的 *section* 后缀留出位置。安装 **/usr/share/man/*** 目录中的文件时通常根据需要将文件名截断至 11 个字符，使得该名称加上三字符的节后缀不超过短文件名系统上的最大文件名长度。

如果未指定 *section* （请参阅前面的参数说明），则 **man** 会按以下顺序搜索手册的所有节： **man1** 、 **man2** 、 **man1M** 、 **man3** 、 **man4** 、 **man5** 、 **man6** 、 **man7** 、 **man8** 、 **man9** 、 **manlocal** 、 **mannew** 、 **manold** ，然后是 **manpublic** ；并输出它遇到的第一个匹配条目。

如果在各节中间存在多个手册条目，则显示第一个手册条目。例如， **man intro** 将仅显示 **intro(1)** 。 **man 4 intro** 将显示 **intro(4)** 。

如果标准输出是电传机，且没有指定 **-** 标志，则 **man** 会利用管道通过 **more** 显示其输出（请参阅 *more*(1) ），如果使用了 **-s** 选项，则消除多个空白行并在显示每一整屏后暂停。通过在用户环境中设置 **PAGER** 变量，可以更改此缺省行为。 **PAGER** 的值必须是命名输出过滤器的字符串（如 *pg*(1) ），以及所需选项。

文件搜索约定

man 根据需要在几个目录中搜索指定的手册条目。搜索会一直进行到找到条目或搜索了所有相关目录为止。依次搜索的前三个目录是： **/usr/share/man** 、 **/usr/contrib/man** 和 **/usr/local/man** 。

可以使用 **MANPATH** 环境变量指定要搜索的目录，如果设置了该环境变量，它将覆盖上面给出的缺省路径。在登录时， **/etc/profile** （或 **/etc/csh.login** ）会将 **MANPATH** 环境变量设置为缺省设置。如果存在文件 **/etc/MANPATH** ，则采用此文件中的缺省设置。 **MANPATH** 变量遵循与 **PATH** 变量相同的形式（请参阅 *environ*(5) ）。

对于其中每个目录， **man** 在 **cat*.Z** 子目录、 **man*.Z** 子目录、 **cat*** 子目录和 **man*** 子目录中进行搜索。 **man*.Z** 和 **man*** 目录中包含条目的与 *nroff*(1) 兼容的源文本。 **cat*.Z** 和 **cat*** 目录中包含条目的已设置格式的版本。 **man*.Z** 和 **cat.Z** 目录中包含压缩格式的条目。在处理这些目录中的文件以便输出或显示之前， **uncompress** 将解压缩（请参阅 *compress*(1) ）这些文件。

如果将 **LANG** 环境变量设置为由 *lang*(5) 定义的任何有效语言名称，且不设置 **MANPATH** 变量或将其设置为缺省目录，则 **man** 会在 **/usr/share/man** 中搜索之前在三个其他目录中搜索手册条目。首先， **man** 在 **/usr/share/man/$LANG** 中搜索，然后在 **/usr/contrib/man/$LANG** 中搜索，最后在 **/usr/local/man/$LANG** 中搜索。因此，如果存在本地语言手册条目并且已正确安装在系统中，则将显示这些条目。

如果将 **MANPATH** 环境变量设置为除缺省值之外的任何值，则不会自动搜索上述以 **$LANG** 作为路径一部分的目录。必须在 **MANPATH** 中明确指定所有目录。 **%L** 、 **%l** 、 **%t** 和 **%c** 说明符可用作路径的组成部分，用于搜索特定语言环境的目录。有关 **MANPATH**的完整说明，请参阅 *environ*(5) 。

man 使用它在所搜索的子目录中找到的最新版本。如果最新版本位于:

man*.Z 将条目解压缩，设置其格式并显示它。如果存在 **cat*.Z** 目录，则会压缩已设置格式的条目，并在 **cat*.Z** 中安装它。如果存在 **cat*** 目录，则会在 **cat*** 中安装已设置格式的条目。

cat*.Z 解压缩并显示条目。

man* 设置条目的格式并显示它。如果存在 **cat*.Z** 目录，则压缩该条目，并将其安装在 **cat*.Z** 中。如果存在 **cat*** 目录，则在 **cat*** 中安装已设置格式的条目。

cat* 显示条目。

如果仅存在 **cat*** 或 **cat*.Z** 子目录且（或）未安装 *nroff*(1) ，则只能显示已设置格式的条目。

如果您选择将已设置格式的条目保留在系统上，请使用缺省设置运行 *catman*(1M) ，这样会创建 **cat*.Z** 目录（在删除系统上存在的任何 **cat*** 目录后），还会创建 **man -k** 选项使用的文件 **/usr/share/lib/whatis** 。如果选择保留 **cat*** 目录，则删除系统上可能存在的任何 **cat*.Z** 目录可以节省空间。请注意，如果同时存在两个目录（ cat* 和 **cat*.Z** ） **man** 将更新这两个目录。

专用手册条目

在某些情况下，可能需要创建手册条目，以供本地使用或由第三方软件供应商进行分发。已经构建了用于设置手册格式的宏，来重新定义页脚，使得不是源自 HP 的手册条目不会在页脚中显示 **HP** 名称。有关此更改的详细信息以及用于 **nroff** 或 **troff** 的手册格式设置宏的说明，请参阅 *man*(5) 。

外部语言环境影响

环境变量

LANG 用于确定显示消息的语言。 **LANG** 还用于确定搜索路径（如上所述）。

如果未指定 **LANG** 或者将其设置为空字符串，则会将缺省值“C”（请参阅 *lang*(5) ）而非 **LANG** 用于消息，但是不用于搜索路径。

如果任一国际化变量中包含无效设置，则 **man** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

如果设置了 **MANPATH** ，则它将提供要在其中搜索给定条目的目录列表，并替换缺省路径。

如果设置了 **PAGER** ，则它定义一个输出过滤器，将使用该过滤器而不是 *more*(1) 对输出进行分页。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

列出在各自的单行说明 （名称）行中包含 **grep** 一词的手册条目：

man -k grep

输出如下所示：

```
grep, egrep, fgrep (1) - search a file for a pattern
zgrep(1)             - search possibly compressed files for a
                       regular expression
```

输出 *grep*(1) 手册条目的单行说明：

man -f grep

输出整个 *grep*(1) 手册条目：

man grep

设置包含紧接在当前目录之下的路径的搜索路径。假定手册条目 **mypage** 存在于目录 **./man1** （或者 **./man1.Z** 、 **cat1** 或 **cat1.Z** ）中。

```
MANPATH=.:/usr/share/man:/usr/contrib/man:/usr/local/man
export MANPATH
man mypage
```

显示 *id*(1) 的手册条目，并通过 **pg -c** 传输输出：

```
PAGER="pg -c"
export PAGER
man id
```

列出为当前系统提供的所有印刷手册（请参阅 *manuals*(5) ）：

man manuals

显示 *intro*(4) 和 *intro*(3) ：

man 4 intro

man 3 intro

警告

构建手册条目时，应该使这些条目可以在照相排字机、传统行式打印机和屏幕显示设备上输出。但是，由于行式打印机和显示设备的限制，在某些情况下可能会造成部分信息的丢失。

文件

/usr/share/lib/whatis	关键字数据库
/usr/share/man/cat*[.Z]/*	已设置格式的手册条目 [已压缩]
/usr/share/man/man*[.Z]/*	原始 (*nroff*(1) 源）手册条目 [已压缩]
/usr/contrib/man/cat*[.Z]/*	
/usr/contrib/man/man*[.Z]/*	
/usr/local/man/cat*[.Z]/*	
/usr/local/man/man*[.Z]/*	
/usr/share/man/$LANG/cat*[.Z]/*	已设置格式的本地语言手册条目 [已压缩]
/usr/share/man/$LANG/man*[.Z]/*	原始 （*nroff*(1) 源）本地语言手册条目 [已压缩]
/usr/contrib/man/$LANG/cat*[.Z]/*	
/usr/contrib/man/$LANG/man*[.Z]/*	
/usr/local/man/$LANG/cat*[.Z]/*	
/usr/local/man/$LANG/man*[.Z]/*	

另请参阅

col(1)、 compress(1)、 grep(1)、more(1)、 catman(1M)、 fixman(1M)、 environ(5)、 intro(1)、 intro(1M)、 intro(2)、 intro(3)、 intro(4)、 intro(5)、 intro(7)、 intro(9)、 introduction(9)、 man(5)、 manuals(5)。

符合的标准

man： XPG4

名称

mediainit - 初始化磁盘或分区 DDS 磁带

概要

mediainit [**-vr**] [**-f** *fmt_optn*] [**-i** *interleave*] [**-p** *size*] *pathname*

说明

mediainit 通过格式化介质、读写测试模式以验证介质的完整性、然后隔离任何所发现的有缺陷的块，来初始化海量存储介质。这个过程可以准备好磁盘或磁带来进行无错误的操作。初始化将破坏正在初始化的区域中所有现有的用户数据。

mediainit 也可以用来分区 DDS 磁带介质。有关更多的详细信息，请参阅下面的 **-p** 选项。

选项

可以识别下列命令选项。可以按任意顺序指定这些选项，但是所有的选项都必须在 *pathname* 之前。不带参数的选项可以单独被列出或被组合在一起。带有参数的选项必须被单独列出，但是选项和其参数之间的空格是任意的。

-v 通常， **mediainit** 仅提供致命的错误消息，并将其直接发送到标准错误中。 **-v** （详细）选项将与 **mediainit** 的低级操作有关的特定设备的信息发送到标准输出 (stdout) 中。该选项对训练有素的服务人员来说是非常有用的，因为它通常需要这些人员有详细的关于设备操作的知识，才能正确地理解这些信息。

-r （重新验证）该选项会强制进行完整的磁带验证，而不论磁带是否已在之前经过了验证。所有以前隔离的块的记录都已被忽略，所以必须重新发现任何坏块。该选项仅在下列情况下使用：

- 磁带上的大量不应被隔离的块怀疑已经被隔离，或
- 有必要破坏（覆盖）磁带上的所有以前的数据。

-f *fmt_optn* 格式选项是范围为 **0** 到 **239** 的特定设备的编号。该选项仅适用于某些支持多种介质格式的 SS/80 设备（独立于交错比）。例如，某些支持 256、512 和 1024 字节的扇区的微型软磁盘驱动器。 **mediainit** 将任何所给出的格式选项直接传送到设备中。如果设备支持该格式选项，则将接受它；如果设备不支持该格式选项，则将拒绝它。有关更多的信息，请参考设备操作手册。缺省的格式选项为 **0** 。

-i *interleave* 交错比， *interleave* 是指顺序逻辑记录和顺序物理记录之间的关系。它定义位于两个连续编号的逻辑记录的开始点之间的介质上的物理记录数。交错比的选择可对磁盘性能产生很大的影响。

-p *size* 将 DDS 盒式磁带介质分区为两个单独的逻辑卷：分区 0 和分区 1：

- *size* 指定分区 1 的最小空间（以 MB 计）。所允许的最大值为 1200。
- 分区 0 为磁带的剩余空间（分区 0 在物理上处于磁带上的分区 1 之后）。

分区 1 的实际大小在某种程度上大于所请求的大小，这样可以在写入时留出磁带介质错误的空间。这样， *size* 为 400 时，将 DDS 磁带格式化为两个分区，其中分区 1 占有至少 400 MB 的数据，而磁带的剩余空间用于分区 0 （对于 1300 MB DDS 盒式磁带，这意味着分区 0 的大小在某种程度上小于 900 MB）。

请注意，除非正在对磁带进行分区，否则没有必要在使用之前格式化 DDS 磁带。未格式化的 DDS 介质在作为单分区磁带使用时不需要初始化。访问单分区磁带上的分区 1 将产生错误。要将两个分区的磁带更改为单分区，请使用 **mediainit** ，同时将 **0** 指定给 *size* 。

pathname *pathname* 指与要被初始化的卷或设备单元相关的字符（原始）设备专用文件的路径名。如果您缺少该设备专用文件的读权限或写权限，或该设备当前对其他任一进程处于打开状态，则 **mediainit** 将异常中止。这样可防止对根设备或任何已安装的卷进行意外地初始化。 **mediainit** 可锁定正在被初始化的单元或卷以防止其他进程进行访问。

除 SCSI 设备以外， *pathname* 必须是设备专用文件，其正在被初始化的设备的最小编号已设置了诊断位。对于设置了诊断位的设备专用文件而言，扇区编号是没有意义的。可访问整个设备。

当给定的单元如设备控制器所定义的那样包含多个卷时，任何与该控制器有关的可用单元或卷，都会独立于共享同一控制器的其他单元和卷而被初始化。这样，您就可以将某一单元或卷初始化为任何格式或交错比，而不影响同类单元或卷上的格式或数据。但是请注意，是整个单元或卷（正如设备控制器所定义的那样）被初始化，而不考虑是否可由操作软件细分成更小的结构。当存在这种结构时，数据可能会意外丢失。

mediainit 可以通过与共享同一控制器的其他单元或卷争用进程，来支配控制器资源并限制访问权限。如果其他同步进程需要访问同一个控制器，则在初始化完成前可以期望降低某些访问权限；如果您正在初始化的盒式磁带位于一个共享根磁盘控制器的驱动器中，则尤其如此。

通常， **mediainit** 试图对任何给出的 **-f** （格式选项）或 **-i** （交错选项）进行认真的检查，并当选项超出范围或不适用于正在进行初始化的介质时异常中止。将交错比或格式选项值指定为 **0** ，与根本不指定选项具有相同的结果。

对于支持交错比的磁盘，可接受的范围通常为 **1** （无交错）到 $n-1$ ，其中 *n* 指每个磁道中的扇区的数目。有关建议的值，请参考相应的设备操作手册。

如果正在初始化的磁盘需要但却不具有指定的交错比，则 **mediainit** 将提供一个适当的却不必是最佳缺省的交错比。

如果给定的设备支持格式选项，则所允许的交错比的范围可能与指定的格式选项有关。在这种情况下，如果已指定了一个交错比，则 **mediainit** 不能对其进行检查。

注释

大多数类型的海量存储介质必须在使用前进行初始化。 HP 硬盘、软盘和盒式磁带需要某些格式的初始化，但是 9 轨磁带不需要初始化。初始化一般包括格式化介质、读写测试模式，然后隔离任何有缺陷的块。根据介质和设备类型的不同，在出厂时可能已进行了部分或全部初始化过程，或者没有进行任何初始化过程。 **mediainit** 会完成

所有适当的步骤来准备好介质，以免发生错误操作。

大多数 HP 硬盘在出厂前通过使用比 **mediainit** 更彻底但也更加耗时的方法进行格式化和详尽的测试。但是，**mediainit** 仍然是非常有价值的，它可以用来在工厂发货之后确保介质的完整性，用来使用正确的交错比进行格式化，并用来隔离自原工厂执行测试以来，任何可能已变成有缺陷的块。

通常， HP 软盘在发货之前不进行格式化，所以它们在使用前必须经过完整的初始化处理。

在验证磁带后，磁带会被彻底测试并且将隔离有缺陷的块。通常仅当先前没有验证过磁带时， **mediainit** 才验证磁带。如果磁带已经经过验证和隔离，则 **mediainit** 通常会重新组织磁带的隔离块表，保留任何以前的隔离，并优化这些隔离的分配来获得连续访问下的最佳性能。重新组织隔离块表只需要几秒钟的时间，然而对于 150 英尺的磁带进行完全验证需要半个小时，对于 600 英尺的磁带则需要一个多小时。

对磁带隔离块表的重新组织在技术上会呈现任何现有的未定义数据，但是通常情况下，该数据不会因覆盖而破坏。为了确保破坏旧的磁带数据（出于安全考虑，这很重要），可以使用 **-r** 选项强制对磁带进行完整的重新验证。

某些应用程序可能需要在使用文件系统前将其放置在介质中。 **mediainit** 不会创建文件系统；它只是准备用于读和写的介质。如果需要此类文件系统，则在运行 **mediainit** 之后，必须调用其他实用程序，例如 **newfs** 、 **lifinit** 或 **mkfs** （请参阅 *newfs*(1M) 、 *lifinit*(1) 和 *mkfs*(1M) ） 。

返回值

mediainit 返回下列值之一:

- **0** 成功完成。
- **1** 产生设备相关错误。
- **2** 遇到语法相关错误。

错误

在 **mediainit** 的执行过程中，相应的错误消息将被输出到标准错误中。

警告

对于在单个控制器上包含多个单元的设备，每个单元都可以独立于任何其他单元而被初始化。但是，应该注意，**mediainit** 要求在初始化开始之前没有任何其他进程访问该设备，而不论要初始化的是哪个单元。如果当前有正在进行的访问，则 **mediainit** 将异常中止。

mediainit 的异常中止可能会使介质处于被破坏的状态，即使该介质在之前已经进行了初始化，也会如此。为了恢复这种被破坏的状态，必须重新启动初始化。

在初始化过程中， **open()** 拒绝对正在初始化的设备进行所有其他的访问，并会产生错误 EACCES （请参阅 *open*(2) ） 。

相关内容

800 系列

不支持 DDS 磁带介质分区（ -p 选项）。

作者

mediainit 由 HP 开发。

另请参阅

lifinit(1)、 mkfs(1M)、 newfs(1M)。

名称

merge - 三方文件合并

概要

merge [**-p**] *file1 file2 file3*

说明

merge 合并作为一个原始文件修订版的两个文件。原始文件是 *file2* ，修订文件是 *file1* 和 *file3* 。 **merge** 可识别从 *file2* 到 *file3* 、从 *file2* 到 *file1* 的所有更改，然后将合并文本存放到 *file1* 中。如果使用了 **-p** 选项，则结果会输出至标准输出，而不是 *file1* 。

如果 *file1* 和 *file3* 在相同位置发生了更改，则会发生重叠。 **merge** 会输出重叠发生的数目，并在结果中同时包含这两个文件中的这一更改。两个文件中发生重叠的更改按以下方式分隔：

> **<<<<<<< file1**
> *file1* 中的行
> **=======**
> *file3* 中的行
> **>>>>>>> file3**

如果有重叠，请编辑 *file1* 中的结果，并删除重叠的更改之一。

此命令对修订控制特别有用，尤其是当 *file1* 和 *file3* 都是以 *file2* 为共同祖先的两个分支端时。

举例

merge 的典型应用如下所示：

1. 要将 RCS 分支合并到主干，请先签出源自 RCS 的三个不同版本（请参阅 *co*(1) ），然后用它们的版本号重命名：5.2、5.11 和 5.2.3.3。文件 5.2.3.3 是文件 5.2 从主干分离出来的 RCS 分支端。
2. 对于本例，假定文件 5.11 是主干上的最新版本，同时也是“原始”文件 5.2 的修订版。则使用以下命令将分支合并到主干：

   merge 5.11 5.2 5.2.3.3

3. 文件 5.11 现在包含分支和主干上的所有更改，并在文件中有显示所有重叠更改的标记。
4. 编辑文件 5.11 以更正重叠部分，然后使用 **ci** 命令重新签入文件（请参阅 *ci*(1) ）。

警告

merge 使用 *ed*(1) 系统编辑器。因此， *ed*(1) 的文件大小限制适用于 **merge** 。

作者

merge 由 Walter F. Tichy 开发。

另请参阅

diff3(1)、 diff(1)、 rcsmerge(1)、 co(1)。

名称

mesg - 允许或拒绝将消息发送到终端

概要

mesg

mesg [**[-]g**] [**[-]n**] [**[-]y**]

说明

命令格式 **mesg [-]n** 通过取消在用户终端上没有适当权限的用户的写入权限，来禁止通过 **write** 发送消息（请参阅 *write*(1) ）。

命令格式 **mesg [-]g** 重新授予相应权限，使其他用户只能使用合法命令（例如 **write** ）来发送消息。

命令格式 **mesg [-]y** 允许使用应用程序（如 **write** 或 **talk** ）将消息发送到用户的终端（即没有任何限制）。

不带任何其他参数的 **mesg** 将报告当前的状态而不会改变该状态。

返回值

mesg 可返回下列值:

- **0** 消息是可接收的。
- **1** 消息是不可接收的。
- **2** 发生了错误。

外部语言环境影响

环境变量

LC_MESSAGES 用于确定显示消息的语言。

如果在环境中未指定 **LC_MESSAGES** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5) ）而非 **LANG**。

如果任一国际化变量中包含无效设置，则 **mesg** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

文件

/dev/tty*

另请参阅

write(1)。

符合的标准

mesg：SVID2、SVID3、XPG2、XPG3、XPG4

名称

mkdir - 创建目录

概要

mkdir [**-p**] [**-m** *mode*] *dirname* ...

说明

mkdir 在模式 0777 下创建指定目录，除非由 **-m** *mode* 选项另外指定，否则该模式可能被 **umask** 更改（请参阅 *umask*(1)）。标准条目 **.**（表示目录本身）和 **..**（表示其父目录）是自动创建的。如果 *dirname* 已存在，则 **mkdir** 将退出并显示诊断消息，而且不更改目录。

选项

mkdir 可识别下列命令行选项：

-m *mode* 如指定的那样创建目录后，文件权限被设置为 *mode* （它是一个为 **chmod** 定义的符号模式字符串）（请参阅 *chmod*(1)）。 **-m** 优先级高于 **umask(1)**。

-p 根据需要创建中间目录。否则， *dirname* 的完整路径前缀必须已存在。 **mkdir** 需要父目录的写入权限。

对于 *dirname* 参数的路径名前缀中的每个目录名，如果它不是现有目录的名称，则指定的目录是使用当前的 **umask** 设置创建的，但是会在每个组件上执行与 **chmod u+wx** 等效的操作以确保 **mkdir** 可以创建更低级别的目录，而不管 **umask** 的设置。将忽略 *dirname* 参数的路径名前缀中与现有目录匹配的每个目录名，且不产生错误。如果中间路径组件存在，但将权限设置为阻止写入或搜索，则 **mkdir** 将失败并显示错误消息。

如果使用 **-m** 选项，则 *dirname* 指定的目录（不包括路径名前缀中的目录）是使用 *mode* 指定的权限创建的。

只能创建 **LINK_MAX** 子目录（请参阅 *limits*(5)）。

访问控制列表 - 仅适用于 JFS 文件系统

如果父目录具有访问控制列表（ACL，请参阅 *aclv*(5)），并且该 ACL 包含缺省条目，则将为新目录创建 ACL，并且父目录的缺省条目将同时作为常规条目和缺省条目应用于新目录的 ACL。

外部语言环境影响

环境变量

LANG 为没有设置或设置为 null（空）的国际化变量提供了缺省值。如果 **LANG** 未设置或设置为 null（空），则会使用缺省值“C”（请参阅 *lang*(5)）。如果任一国际化变量中包含无效设置，则 **mkdir** 的行为类似于所有国际化变量都设为“C”。请参阅 *environ*(5)。

LC_ALL 如果设为非空字符串值，则会覆盖所有其他国际化变量的值。

LC_CTYPE 用于确定将文本解释为单字节字符和（或）多字节字符，将字符的类别解释为可输出，以及确定与正则表达式中的字符类表达式相匹配的字符。

LC_MESSAGES 确定语言环境，该语言环境应该用于影响写入到标准错误的诊断消息以及写入到标准输出的信

息消息的格式和内容。

NLSPATH 用于确定消息清单的位置，以便处理 **LC_MESSAGES** 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

诊断信息

mkdir 如果已成功创建所有目录，则返回退出代码 0。否则将输出诊断信息并返回非零值。

如果指定了 **-p** 选项，并且所有指定目录现在都存在，则 **mkdir** 将返回退出代码 0。如果任一中间目录没有搜索或写入权限（通过 **-p** 选项来授予），则 **mkdir** 将输出诊断信息并返回非零值。

举例

在当前目录中的现有目录 **raw** 下创建目录 **gem**：

mkdir raw/gem

在当前目录下创建目录路径 **raw/gem/diamond** ，并将目录 **diamond** 的权限设置为对所有用户都是只读的 (**a=r**)：

mkdir -p -m "a=r" raw/gem/diamond

它等效于（请参阅 *chmod*(1) ）：

mkdir -p -m 444 raw/gem/diamond

如果目录 **raw** 或 **raw** 和 **gem** 已存在，则仅创建指定路径中缺少的目录。

另请参阅

rm(1)、 setacl(1)、 sh(1)、 umask(1)、 aclv(5)。

符合的标准

mkdir：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2

名称

mkfifo - 创建 FIFO（命名管道）专用文件

概要

mkfifo [**-p**] [**-m** *mode*] *filename* ...

说明

mkfifo 创建根据其操作数列表命名的 FIFO 专用文件。该命令按照指定的顺序使用这些操作数，如果用户具有对相应目录的写入权限，则会使用用户的文件模式创建掩码（请参阅 *umask*(2) ）所修改的权限 0666 来创建 FIFO 。

执行的具体操作等同于对操作数列表中的每个文件名执行以下调用：

mkfifo(*filename***, 0666)**

（请参阅 *mkfifo*(3C) ）。

选项

mkfifo 可识别下列命令行选项：

-m *mode* 在创建 FIFO 专用文件后，将新文件的权限位设置为指定的 *mode* 值。 *mode* 选项参数是符号模式字符串，如 *chmod*(1) 中所定义。

（仅适用于 XPG4）在符号模式字符串中，运算符 **+** 和 **-** 的解释是以 a=rw 的初始模式为基准。

-p 创建所需的中间路径名部分。

外部语言环境影响

环境变量

LANG 确定当 **LC_ALL** 和相应的环境变量（以 **LC_** 开头）都未指定语言环境时，语言环境类别将使用的语言环境。如果未指定 **LANG** 或将其设置为空字符串，则使用缺省值“C”（请参阅 *lang*(5) ）。

LC_ALL 确定用于覆盖各语言环境类别的任何值（通过设置 **LANG** 或以 **LC_** 开头的任何环境变量来指定）的语言环境。

LC_CTYPE 确定将区域设置文本字节序列解释为何种字符（例如参数中的单字节和多字节字符）。

如果任一国际化变量中包含无效设置，则 **mkfifo** 认为所有国际化变量都被设为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集。

返回值

如果用至少一个操作数调用 **mkfifo** 命令，并成功创建了所有 FIFO 专用文件，则该命令将返回零。否则会输出一条诊断消息并返回非零值。

举例

以下命令将在当前目录中创建一个名为 **peacepipe** 的 FIFO 专用文件：

mkfifo peacepipe

另请参阅

chmod(1)、 umask(1)、 mknod(1M)、 mkfifo(3C)。

符合的标准

mkfifo：XPG3、XPG4、POSIX.2

名称

mkmf - 创建生成文件

概要

mkmf [**-acdeil**] [**-f** *makefile*] [**-F** *template*] [**-M** *language*] [*macroname***=***value* ...]

说明

mkmf 命令创建一个生成文件，该文件会通知 **make** 命令如何构建和维护程序和库（请参阅 *make*(1) ）。在当前工作目录中收集完所有源代码文件名并将其插入到生成文件之后， **mkmf** 将扫描源代码文件以查找包含的文件，然后生成要追加到生成文件的相关性信息。源代码文件是通过它们的文件名后缀来标识的。 **mkmf** 可识别下列后缀：

.c	C
.B	C++
.f	FORTRAN
.h	包含文件
.i	Pascal 包含文件
.l	Lex 或 Lisp
.o	对象文件
.p	Pascal
.r	Ratfor
.s	汇编程序
.y	Yacc

mkmf 命令在创建生成文件之前先检查现有的生成文件。如果 **-f** 选项不存在，则 **mkmf** 将尝试分别创建 **makefile** 和 **Makefile** 。

在创建生成文件之后，可以使用文本编辑器进行任意更改。也可以使用 **mkmf** 重新编辑生成文件中的宏定义，而无论自创建以来是否进行了更改。

缺省情况下， **mkmf** 将创建一个程序生成文件。要创建一个处理库的生成文件，必须使用 **-l** 选项。

创建请求

假定一个由 **mkmf** 创建的生成文件， **make** 可识别下列请求：

all	编译并加载程序或库。
clean	删除所有的对象文件和核心文件。
clobber	删除所有可重新生成的文件。
depend	更新生成文件中包含的文件相关性。
echo	在标准输出中列出源代码文件的名称。

extract 从库中提取所有对象文件，并将它们放在与源代码文件相同的目录下，但不更改库。

index 在标准输出中输出功能索引。

install 编译并加载程序或库，并将其移到它的目标目录下。

print 在标准输出中输出源代码文件。

tags 为 **ex** 编辑器（请参阅 *ex*(1) 和 *ctags*(1) ）以及 C、Pascal 和 Fortran 源代码文件创建标记文件。

update 仅当存在比程序或库更新的源代码文件时，才重新编译，然后链接和安装程序或库。

可以同时给出多个请求。例如，要 (1) 编译和链接程序， (2) 将程序移到其目标目录下， (3) 删除任何不需要的对象文件，请使用：

make install clean

宏定义

mkmf 识别下列宏定义：

CFLAGS C 编译程序标志。在当前处理的目录下搜索包含的文件后， **mkmf** 在使用 **-I** 编译程序选项命名的目录下搜索，然后在 **/usr/include** 目录下搜索。

COMPILESYSTYPE
/usr/include 的位置。如果定义了 **COMPILESYSTYPE** 宏或环境变量，则 **mkmf** 在 **/$COMPILESYSTYPE/usr/include** 而不是 **/usr/include** 中搜索包含的文件。

CXXFLAGS C++ 编译程序标志。在当前处理的目录下搜索包含的文件后， **mkmf** 在使用 **-I** 编译程序选项命名的目录下搜索，然后在 **/usr/include/CC** 目录下搜索，接着在 **/usr/include** 目录下搜索。

DEST 要安装程序或库的目录。

EXTHDRS 位于当前目录外部的包含的文件的列表。如果生成了相关性信息，则 **mkmf** 会自动更新生成文件中的宏定义。

FFLAGS Fortran 编译程序标志。在当前处理的目录下搜索包含的文件后， **mkmf** 在使用 **-I** 编译程序选项命名的目录下搜索，然后在 **/usr/include** 目录下搜索。

HDRS 当前目录下包含的文件的列表。 **mkmf** 自动更新生成文件中的该宏定义。

INSTALL 安装程序名称。

LD 链接编辑器名称。

LDFLAGS 链接编辑器标志。

LIBRARY 库名称。该宏也暗含了 **-l** 选项的功能。

LIBS 链接编辑器所需的、用于解析外部引用的库的列表。

MAKEFILE 生成文件名称。

OBJS 对象文件列表。 **mkmf** 自动更新生成文件中的该宏定义。

PROGRAM 程序名称。

SRCS 源代码文件列表。 **mkmf** 自动更新生成文件中的该宏定义。

SUFFIX 其他文件名后缀的列表，可供 **mkmf** 识别。

SYSHDRS 在 **/usr/include** 目录层次结构中找到的包含的文件的列表。如果生成了相关性信息，则 **mkmf** 自动更新生成文件中的该宏定义。如果生成文件中忽略了 **SYSHDRS** ，则 **mkmf** 不生成 **/usr/include** 相关性信息。

上述这些宏定义以及生成文件中已有的所有其他宏定义，可以替换为命令行中形式为 *macroname***=***value* 的定义。例如，要更改 C 编译程序标志和程序名称，请键入以下命令行：

mkmf "CFLAGS=-I../include -O" PROGRAM=mkmf

请注意，带有空格的宏定义（例如 **CFLAGS** ）必须用双引号 (**"**) 括起。

环境

环境是由 **mkmf** 读取的。除了 **HDRS** 、 **EXTHDRS** 、 **SRCS** 和 **OBJS** ，所有变量都假定为宏定义。环境变量是在命令行宏定义和 *makefile* 中的宏定义之后进行处理的。 **-e** 选项强制环境覆盖 *makefile* 中的宏定义。

文件名后缀

mkmf 可以识别其他文件名后缀，或忽略已经识别的后缀，方法是在 **SUFFIX** 宏定义中指定后缀说明。每个后缀说明均采用 **.***suffix***:***tI* 的形式，其中 *t* 是一个表示文件内容的字符（ s = 源文件， **o** = 对象文件， **h** = 头文件， **x** = 可执行文件）， *I* 是一个可选字符，表示头文件的包含语法（ C = C 语法， **C++** = C 语法并增加 **/usr/include/CC** 作为标准搜索目录， **F** = Fortran 和 Ratfor 语法， **P** = Pascal 语法）。下表显示了 **mkmf** 的缺省配置：

.c:sC	C
.B:sC++	C++
.f:sF	Fortran
.h:h	包含文件
.i:h	Pascal 包含文件
.l:sC	Lex 或 Lisp
.o:o	对象文件
.p:sP	Pascal
.r:sF	Ratfor
.s:s	汇编程序
.y:sC	Yacc

例如，要将对象文件后缀更改为 **.obj** ，则不定义 Pascal 包含文件后缀，并禁止扫描 Fortran 文件以搜索包含的文件， **SUFFIX** 宏定义可以是：

SUFFIX = .obj:o .i: .f:s

Include 语句语法

对于 C、C++、 Fortran 和 Pascal 源代码，include 语句的语法采用下列形式：

C/C++：

#include "*filename***"**
#include *filename*

其中 **#** 必须是行中的第一个字符。

Fortran：

$include ' *filename* **'$**
$INCLUDE ' *filename* **'$**

其中 **$** 必须是行中的第一个字符。另外，如果 include 语句从第 7 列开始，则可以忽略 **$**。在任意一种情况下，都可以忽略末尾的 **$** 。

Pascal：

$include ' *filename* **'$**
$INCLUDE ' *filename* **'$**

其中 **$** 必须是行中的第一个字符，并且末尾的 **$** 是可选的。

用户定义的模板

如果 **mkmf** 在当前目录下找不到生成文件，它通常会使用 **/usr/ccs/lib/mf** 下的标准模板 **C.p** 或 **C.l** 之一，除非用户在目录 **$PROJECT/lib/mf** （其中 **$PROJECT** 是指定给 **PROJECT** 环境变量的目录的绝对路径名）下有备用的 **C.p** 或 **C.l** 模板文件。

选项

mkmf 可识别下列选项：

-a 在生成文件中包括以 **.** 开头的源文件。

-c 禁止显示“**creating** *makefile* **from** ... ”消息。

-d 禁用扫描源代码以查找 **include** 文件的操作。生成文件中的旧的相关性信息保持原状。

-e 环境变量将覆盖 *makefile* 中的宏定义。

-f *makefile* 指定备用的 *makefile* 文件名。缺省文件名为 **Makefile** 。

-i 提示用户输入程序或库的名称以及它的安装目录。如果对于每个查询都键入回车作为响应，则 **mkmf** 会假定缺省程序名为 **a.out** 或缺省库名为 **lib.a** ，并且目标目录为当前目录。

-l 强制生成文件成为一个库生成文件。

-F *template* 指定备用的生成文件模板路径名。该路径名可以是相对路径名，也可以是绝对路径名。

-M *language* 指定备用的 *language* 特定的生成文件模板。缺省语言为 C，对应的程序和库生成文件模板分别是 **C.p** 和 **C.l** 。 **mkmf** 在 **/usr/ccs/lib/mf** 或 **$PROJECT/lib/mf** 中查找这些模板。

诊断信息

退出状态 0 表示正常。退出状态 1 表示出错。

警告

生成文件的名称作为一个宏定义包含在生成文件中，如果重命名生成文件，则必须对它进行更改。

由于可执行文件依赖于库，因此，在生成文件的 **LIBS** 宏定义中，必须将标准库的缩写扩展为完整路径名。

在生成文件中，生成的相关性信息位于以 **###** 开头的那一行之后。不能删除该行，也不能将任何其他信息插入生成文件中该行的下面。

程序名或库名不能与生成文件中任何预定义的目标名称发生冲突。避免名称 **update** 阻止 **make** 无数次递归执行它自身，这一点是尤其重要的。

作者

mkmf 由加州大学伯克利分校开发。

文件

/usr/ccs/lib/mf/C.p	标准程序生成文件模板
/usr/ccs/lib/mf/C.l	标准库生成文件模板
$PROJECT/lib/mf/C.p	用户定义的程序生成文件模板
$PROJECT/lib/mf/C.l	用户定义的库生成文件模板

另请参阅

ar(1)、 ctags(1)、 ld(1)、 make(1)。

《Software-Practice and Experience》中的“Automatic Generation of Make Dependencies”， Walden, K.，第 14 卷，第 6 期，第 575-585 页，1984 年 6 月。

名称

mkmsgs - 创建可由 gettxt() 使用的消息文件

概要

mkmsgs [**-i** *locale*] [**-o**] *textfile msgfile*

说明

mkmsgs 命令使用包含本地化文本字符串的文件作为输入，并生成可以由 *gettxt*(3C) 例行程序访问的消息文件。*textfile* 是包含文本字符串的文件的名称。 *msgfile* 是输出消息文件的名称。 **mkmsgs** 会将后缀 **.cat** 追加到消息文件名后。对于短文件名文件系统，文件名的总长度应该小于 14 字节。 *msgfile* 文件不应该包含冒号，因为它将使格式化例行程序不知所措。

textfile 中包含本地化文本字符串。文本字符串由换行符分隔。文本字符串按顺序进行处理，并被复制到 *msgfile* 中。输入中的空行将导致在 *msgfile* 中写入相应的空消息。

选项

mkmsgs 命令支持下列选项：

-i *locale* *msgfile* 安装在与指定的 *locale* 相对应的系统级的本地化目录中。只有具有适当特权的用户才可以创建或覆盖该目录中的消息文件。如果该目录不存在，则将创建该目录。

-o 如果 *msgfile* 存在，则覆盖该文件。

外部语言环境影响

环境变量

LC_CTYPE 可确定将消息解释为单字节字符和（或）多字节字符。

如果将 **LANG** 设置为一种有效语言，并且可显示 **LANG** 语言的消息，则会以该语言显示消息。否则，将发布“C”语言环境消息。

如果在环境中未指定 **LC_CTYPE** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则使用缺省的“C”（请参阅 *lang*(5) ）而非 **LANG**。如果任一国际化变量包含无效设置，则 **mkmsgs** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

以下示例说明了输入文本字符串的格式：

```
global %s not found\n
\n\n<press return to continue>\n\n
\t%s, %d, %d,typ = %d, disp = '%s'\n
```

警告

mkmsgs 只是为了与 SVID3 兼容而提供的。建议用户使用由 HP 和 X/Open Company, Ltd. 联合开发的 NLS 机制。

另请参阅

gencat(1)、 gettxt(3C)、 setlocale(3C)。

符合的标准

mkmsgs： SVID3

名称

mkstr - 将来自 C 源文件的错误消息提取到一个文件中

概要

mkstr **[-]** *messagefile prefix file* ...

说明

mkstr 可检查一个 C 程序，并创建包含由该程序使用的错误消息字符串的文件。通过引用文件中的位置，可以使有许多错误诊断信息的程序小很多，并可降低运行程序时的系统开销。

mkstr 处理每个指定的 *file* ，将每个文件的修订版本放在一个名称为原始名加指定 *prefix* 的文件中。 *mkstr* 的典型用法是：

mkstr mystrings xx *.c

此命令会将当前目录中 C 源文件中的所有错误消息放在文件 *mystrings* 中，并将这些源文件的修订副本放在由原始名加前缀 *xx* 所指定的文件中。

在处理源文件中的错误消息以便将其传输到消息文件中时， **mkstr** 将在输入文件中搜索字符串 **error(** 。每次遇到该字符串时，都会将从前导引号之后开始的 C 字符串放在消息文件中，后跟一个空字符和换行符。空字符可终止消息，以便在检索时可以轻松地使用它，而换行符可用于方便地列出错误消息文件（使用 **cat** 、 **more** 等 — 请参阅 *cat*(1) 和 *more*(1) ）并查看其内容。

输入文件的已修改副本和原始文件几乎完全相同，不同之处在于，已经移动到错误消息文件中的任何字符串，只要出现就会被替换为偏移指针， **lseek** 用偏移指针检索该消息。

如果命令行包含可选的 **-**，则提取的错误消息将会放置在指定消息文件的结尾处（追加）而不是覆盖它。这样，您可以处理作为以前由 **mkstr** 处理过的大型程序组成部分的各个文件，而无需重新处理所有文件。

对于由原始程序使用的、名称以“error”结尾，且也可以采用常量字符串作为第一个参数的所有函数，应该进行重写以便使其在错误消息文件中搜索字符串。

例如，基于前面的示例用法的程序类似于：

```
#include <stdio.h>
#include <sys/types.h>
#include <fcntl.h>

char errfile[] = "mystrings" ;

error(offset, a2, a3, a4)
int offset, a1, a2, a3;
{
   char msg[256];
   static int fd = -1;

   if (fd < 0) {
```

```
        fd = open(errfile, O_RDONLY);
        if (fd < 0) {
                perror(errfile);
                exit(1);
        }
    }

    if (lseek(fd, (off_t) offset, 0) || read(fd, msg, 256) <= 0) {
            printf("? Can't find error message in %s:\n", errfile);
        perror(errfile);
        exit(1);
    }

    printf(msg, a1, a2, a3);
}
```

外部语言环境影响

环境变量

LC_CTYPE 用于确定将注释和字符串解释为单字节字符和（或）多字节字符。

如果未在环境中指定 **LC_CTYPE** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则会使用缺省值“C”（请参阅 *lang*(5)）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **mkstr** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5)。

国际代码集支持

在文件名、注释和字符串内，支持单字节字符代码集和多字节字符代码集。

另请参阅

lseek(2)、 perror(3C)、 xstr(1)。

缺陷

名称以 **error** 结尾的函数调用中的字符串（典型示例为 **perror()**）可能会被 **mkstr** 替换为偏移量。

对第一个参数不是字符串常量的错误函数的调用不会被修改，且不发出任何警告。

名称

mktemp - 为临时文件创建名称

概要

mktemp [**-c**] [**-d** *directory_name*] [**-p** *prefix*]

说明

mktemp 创建可用于临时文件路径名的名称，并将该名称写入标准输出。选择名称的原则是不与现有文件重名。如果指定了 **-c** 选项，则会创建一个具有生成的文件名的零长度文件。

mktemp 生成的名称由目录名、斜线 (**/**) 、 **LOGNAME** 环境变量的值（截断至 **{NAME_MAX}** – 6 个字符）以及调用进程的进程 ID 连接而成。

目录名的选择方式如下:

1. 如果指定了 **-d** 选项，则使用 *directory_name* 。
2. 否则，如果设置了 **TMPDIR** 环境变量，且可以通过将该变量的值用作目录名来获取可以生成唯一名称的字符串，则使用该值。
3. 否则，如果可以通过将 **/tmp** 用作目录来获得可生成唯一名称的字符串，则使用 **/tmp** 。
4. 否则，使用 **.** （当前目录）。

如果指定了 **-p** 选项，则生成名称时将使用 *prefix* 而非 **LOGNAME** 环境变量值。

返回值

mktemp 成功完成时返回零，如果遇到语法错误、文件访问问题、文件创建错误或无法生成唯一路径名，则返回非零值。

另请参阅

mktemp(3C)、 umask(1)。

名称

mm、osdd - 输出用 mm 宏格式化的文档

概要

mm [*options*] [*files*]

osdd [*options*] [*files*]

说明

通过 **nroff** 和 **mm** 文本格式化宏程序包（请参阅 *nroff*(1) ）， **mm** 可用于格式化并输出文档。该命令的选项可指定由 **tbl** 和（或） **neqn** （请参阅 *tbl*(1) 和 *neqn*(1) ）进行预处理，以及由各种面向终端的输出过滤器进行后处理。根据所选择的选项，会生成适当的管道线以及 **nroff** 和 **mm** 所需的参数和标志。

osdd 相当于命令 **mm -mosd** 。

选项

mm 可识别下列 *options* 和命令行参数。任何其他参数或选项（例如 **-rC3** ）将根据需要传递到 **nroff** 或 **mm** 。这些选项可以按任何顺序排列，但它们必须出现在 *files* 参数之前。如果没有给定任何参数，则 **mm** 将输出其选项的列表。

-T*term* 指定输出终端的类型；对于 *term* 的已识别值的列表，输入 **help term2** 。如果 不使用该选项，则 **mm** 将使用 Shell 环境变量 **$TERM** （请参阅 *profile*(4) 和 *environ*(5) ）的值作为 *term* 的值（如果已设置 **$TERM** ）；否则， **mm** 将使用 **450** 作为 *term* 的值。如果指定了几种终端类型，则将使用最后一种类型。

-12 表示将以 12 间距生成文档。当将 **$TERM** 设置为 **300** 、 **300s** 、 **450** 和 **1620** 之一时，可以使用该选项（如果使用该选项，则必须将 DASI 300 和 300s 终端上的间距选项手动设置为 **12**）。

-c 使 **mm** 调用 *col*(1) ；请注意，如果 *term* 不是 **300** 、 **300s** 、 **450** 、 **37** 、 **4000a** 、 **382** 、 **4014** 、 **tek** 、 **1620** 和 **X** 之一的话，则 **mm** 会自动调用 *col*(1) 。

-e 使 **mm** 调用 **neqn** 。

-t 使 **mm** 调用 **tbl** 。

-E 调用 **nroff** 的 **-e** 选项。

诊断信息

如果所有参数都不是可读文件并且 **mm** 不作为过滤器使用， **mm** 将发送消息 **mm: no input file** 。

举例

假定环境中的 Shell 变量 **$TERM** 已设置为 **450** ，则下面的两个命令行的作用是相等的：

mm -t -rC3 -12 ghh*
tbl ghh* | nroff -cm -T450-12 -h -rC3

当指定了 **-** 时， **mm** 将读取标准输入，而不是任何文件名（注意，如果其他文件与 **-** 一起使用，则会出现灾难性故障）。该选项允许 **mm** 作为过滤器使用，如下面的示例所示：

cat dws | mm -

提示

- **mm** 通过 **-h** 选项调用 **nroff** 。通过该选项， **nroff** 假定终端每隔 8 个字符位置设置 Tab 功能。
- 使用 **nroff** 的 **-o***list* 选项可以指定要输出的页范围。但请注意，如果进行调用时 **mm** 带有一个或多个 **-e** 、 **-t** 和 **-** 选项，同时使用了 **nroff** 的 **-o***list* 选项，则当文档最后一页未在 *list* 中指定时，将生成无害的“断开的管道”诊断信息。
- 如果使用 **nroff** 的 **-s** 选项（来停止页面间的输出），则使用换行符（而不是回车符或换行符）可重新启动输出。 **nroff** 的 **-s** 选项不能与 **mm** 的 **-c** 选项一起使用，也不能在 **mm** 自动调用 **col** 时使用（请参阅上面的 **-c** 选项和 *col*(1) ）。
- 如果指定一个不正确的输出终端类型，则 **mm** 将产生（通常是敏感的）不可预料的结果。但是，如果正在将输出重定向到文件，则可使用 **-T37** 选项，然后在实际输出格式化文件后使用适当的终端过滤器。

另请参阅

col(1)、 env(1)、 nroff(1)、 tbl(1)、 profile(4)、 term(4)、 mm(5)。

名称

model - 输出硬件型号信息

概要

model

说明

model 输出计算机硬件型号。 **model** 还显示制造商、产品名称或其他信息。

举例

下面的 **model** 输出结果表示 Hewlett-Packard 的基于 Itanium(R) 的 g4000 服务器。

ia64 hp server g4000

下面的 **model** 输出结果是 HP 9000 L 系列服务器的型号信息。

9000/800/L1000-36

警告

需要可以跨平台移植的应用程序，不应具有对内部结构或型号字符串内容的依赖性。每种新产品的型号字符串都可以使用独有的格式。下列内容在型号输出结果中可能不同：包括在型号字符串中的信息类型、信息字段的顺序或信息字段之间的分隔符。

另请参阅

getconf(1)、 uname(1)。

名称

more、page - 用于屏幕显示的文件读取过滤器

概要

more [**-**_n_] [**-cdefisuvz**] [**-n** _number_] [**-p** _command_] [**-t** _tagstring_] [**-x** _tabs_] [**-W** _option_] [**+**_linenumber_] [**+/**_pattern_] [_name_]...

page [**-**_n_] [**-cdefisuvz**] [**-n** _number_] [**-p** _command_] [**-t** _tagstring_] [**-x** _tabs_] [**-W** _option_] [**+**_linenumber_] [**+/**_pattern_] [_name_]...

备注

在某些标准中优先使用 **pg** 命令，该命令具有某些附加功能，但它不支持字符突出显示（请参阅 _pg_(1)）。

说明

more 是一种过滤器，用于显示连续文本，可在屏幕终端上一次显示一整屏文本。该命令与 **pg** 很相似，主要为了向后兼容而保留。 **more** 通常在整屏显示后暂停，同时在屏幕底部显示文件名。要显示下一行，可按 **Return** 键。要显示另一屏，可按 **Space** 键。后面将叙述其他可能的情况。

more 和 **page** 仅有微小差别。 **more** 在输出下一页时向上滚屏，而 **page** 在输出新页时会清屏并输出新的整屏文本。两者都在相邻整屏文本之间提供了一个重叠行。

name 可以是文件名或 **-** （指定标准输入）。 **more** 以文件参数的给定顺序处理这些参数。

more 支持基本正则表达式语法（请参阅 _regexp_(5)）。

more 可识别下列命令行选项：

-n _number_　将显示窗口中的行数设为 _number_ ，它是一个十进制正整数。缺省的行数为终端可显示的行数减 1；对于可显示 24 行的屏幕，缺省值为 23。 **-n** 标志将覆盖所有从环境获取的值。

-_n_　除了行数被设置为 _n_ 之外，与 **-n** _number_ 相同。

-c　从屏幕顶部开始显示每一页，并在显示前清除每一行。这样可避免滚动屏幕，使得当 **more** 输出数据时更易于读取。如果终端没有清除至行尾的功能，则该选项会被忽略。

-d　在每一整屏的结束处提示用户： **Press space to continue, q to quit, h for help** 。当在某些情况下（例如在有许多初级用户参加的培训班中）使用 **more** 作为过滤器时，这一选项十分有用。

-e　在输出了参数表中最后一个文件的最后一行后立即退出。

-f　使用逻辑行数而非屏幕行数。即，不折叠较长的行。如果用管道通过 _ul_ 传送 _nroff_ 输出，则推荐使用该选项，这是因为 ul 可生成转义序列。这些转义序列中包含的字符通常会占据屏幕位置，但当作为转义序列的一部分发送至终端时不会输出。因此， **more** 可能会认为行比实际的要长，并错误地折叠行。

-i　在搜索时执行模式匹配，不考虑大小写。

-s　压缩输出中的多个空行，仅生成一个空行。该选项可在屏幕上显示尽可能多的有用信息，在查看 _nroff_ 输出时尤为有用。

-u 通常， **more** 以适合于特定终端的方式处理带下划线及粗体的文本（例如 *nroff* 生成的）：如果终端支持加下划线或具有突出显示（通常为反显）模式，则 **more** 将输出相应转义序列为源文件中带下划线的信息启用下划线或突出显示模式。如果终端支持突出显示功能，则 **more** 使用应该以粗体显示的模式信息。 **-u** 选项禁止这种处理，与“ul”和“os”terminfo 标志一样。

-v 不以图形方式显示非打印字符；缺省情况下，几乎所有的非 ASCII 字符和控制字符（除 **Tab** 、 **Backspace** 、和 **Return**) 之外）都以可视形式显示： ˆ*x* 代表 **Ctrl-** *x* 、或 **M-***x* 代表非 ASCII 字符 *x* 。

-z 与不指定 **-v** 基本相同，但将 **Backspace** 显示为 **ˆH** 、将 **Return** 显示为 **ˆM** ，并将 **Tab** 显示为 **ˆI** 。

-p *command* 开始时在检查的每个文件的 *command* 参数中执行 **more** 。如果该命令是定位命令，例如行号或正则表达式搜索，则将当前位置设置为表示该命令的最终结果，而不会输出文件的任何中间行。如果定位命令不成功，则文件的第一行为当前位置。

-t *tagstring* 整屏输出包含以 *tagstring* 命名的标记的文件。指定的标记出现在当前位置。如果同时指定了 **-p** 和 **-t** 选项，则 **more** 首先处理 **-t** ；即通过 **-t** 选择包含 *tagstring* 的文件，然后再执行该命令。

-x *tabs* 在每个 *tabs* 位置设置制表位。 *tabs* 参数的缺省值为 8。

-W *option* 为 **more** 命令提供可选扩展。目前，支持以下两个选项：

notite 防止 **more** 在显示文件之前发送终端初始化字符串。该参数还可防止 **more** 在退出之前发送终端取消初始化字符串。

tite 使 **more** 发送初始化和取消初始化字符串。这是缺省值。

+*linenumber* 开始列出时使当前位置设置为 *linenumber* 。

+/*pattern* 开始列出时使当前位置设置为与正则表达式 *pattern* 匹配的行之上的两个行。

注释：与编辑器不同，该结构不能以 **/** 结束。如果以斜线结束，则在搜索模式下会将末尾的斜线视为字符。

每屏可显示的行数由选项 **-n** （如果存在）或通过检查环境值确定。输出的实际行数比该数少一行，这是因为屏幕的最后一行用于输出用户提示和用户输入。

每行可显示的列数通过检查环境中的值来确定。 **more** 可输出包含多于该列数的字符的行，方法是将一个行分解成多个逻辑行，在这些行中，除最后一行外，其他每一行都包含填充所有列所需的字符。逻辑行独立于彼此输出；即影响某一行的命令不会影响其他行。

在确定行数和列数时，如果上述方法不生成任何数值，则 **more** 将使用 terminfo 描述符文件（请参阅 *term*(4) ）。如果这种方法也不成功，则会将行数设置为 24，将列数设置为 80。

如果标准输出为终端且未指定 **-u** ，则 **more** 将以特殊方式处理退格字符和回车符。

- 一个字符后面跟一个退格字符，然后再后接一个下划线 (_) 时，如果终端支持下划线，则会将该字符输出为为带下划线的文本。一个下划线后面跟一个退格字符，然后再后接任一字符时，如果终端支持下划线，也会将该字符输出为带下划线的文本。
- 对于出现在两个相同可打印字符之间的退格字符，如果终端类型支持粗体形式，则会以粗体形式输出这两个字符中的第一个字符，并忽略第二个字符。紧接着出现的后续退格/字符（与第二个字符相同）对同样会被丢弃。
- 其他退格字符序列被直接输出到终端，这通常会使退格字符前的字符不会显示出来。
- 忽略行末的回车字符，而不是将其输出为控制字符。

如果标准输出不是终端设备，则 **more** 始终会在到达参数列表中最后一个文件的末尾时退出。否则，对于除最后一个文件以外的其他所有文件， **more** 将提示已达到文件末尾，并提示下一个文件的名称。对于指定的最后一个文件或标准输入（如果未指定文件）， **more** 将提示到达文件末尾并接受其他命令。如果下一个命令指定向前滚动，则 **more** 将退出。如果指定了 **-e** 选项， **more** 将在输出最后一个文件的最后一行后立即退出。

more 使用环境变量 **MORE** 来预设所需的任何标志。因此， **MORE** 变量可设置一个字符串，其中包含标志和参数，前面有连字符并用空白字符分隔，类似于命令行的标志和参数。任何命令行标志或参数都在 **MORE** 变量的标志或参数之后进行处理，类似于以下命令行：

more $MORE *flags arguments*

例如，要使用 **-c** 操作模式查看文件，Shell 命令序列：

MORE='-c' ; export MORE

或 *csh* 命令：

setenv MORE -c

会使所有的 **more** 调用，包括由诸如 *man* 和 *msgs* 等程序进行的调用，都使用该模式。设置 **MORE** 环境变量的命令序列通常位于 *.profile* 或 *.cshrc* 文件中。

在下面的说明中， 当前位置指以下两种位置：

- 屏幕上当前行的位置
- 屏幕上当前行在文件中的行号

与当前位置相对应的屏幕上的行是第三行。如果这是不可能的（将显示的内容少于三行，或者是文件的第一页或最后一页），则当前位置为屏幕上的第一行或最后一行。

当 **more** 暂停时可键入的其他序列及其作用如下（ i 为可选整数参数，缺省为 1）：

*i***Return**
*i***j**
*i***Ctrl-e**

***i*Space** 向前滚动 *i* 行。 **Space** 的缺省 *i* 为一整屏；对于 **j** 和 **Return** 为一行。即使 *i* 大于屏幕大小，也会输出所有 *i* 行。在文件末尾， **Return** 可使 **more** 继续显示列表中的下一个文件，或者当前文件为列表中的最后一个文件时退出。

***i*d**
***i*Ctrl-d** 向前滚动 *i* 行，缺省为屏幕大小的一半。如果指定了 *i* ，则它将成为后续 **d** 和 **u** 命令的新的缺省值。

***i*u**
***i*Ctrl-u** 向后滚动 *i* 行，缺省为屏幕大小的一半。如果指定了 *i* ，则它将成为后续 **d** 和 **u** 命令的新的缺省值。

***i*k**
***i*Ctrl-y** 向后滚动 *i* 行，缺省为一行。即使 *i* 大于屏幕大小，也会输出所有 *i* 行。

***i*z** 继续显示 *i* 行，并将新窗口（整屏）大小设置为 *i* 。

***i*g** 转至文件中的第 *i* 行，缺省为 1（文件开头）。滚动或重写屏幕，使该行位于当前位置。如果未指定 *i* ，则 **more** 将显示文件的第一个整屏。

***i*G** 转至文件中的第 *i* 行，缺省为文件末尾。如果未指定 *i* ，则会滚动或重写屏幕，使文件的最后一行位于屏幕最底部。如果指定了 *i* ，则会滚动或重写屏幕，使该行位于当前位置。

***i*s** 向前跳过 *i* 行，缺省为 1，并从该点开始输出下一整屏文本。如果 *i* 对应的当前位置会导致输出少于一个整屏的文本，则将输出文件中的最后一整屏文本。

***i*f**
***i*Ctrl-f** 向前移动 *i* 行，缺省为一整屏。在文件末尾， **more** 将继续显示列表中的下一个文件，如果当前文件是列表中的最后一个文件则退出。

***i*b**
***i*Ctrl-b** 向后移动 *i* 行，缺省为一整屏。如果 *i* 大于屏幕大小，则仅输出最后一整屏内容。

q
Q
:q
:Q
ZZ 退出 **more** 。

=
:f
Ctrl-g 输出当前正在检查的文件的名称、相对于待检查文件总数的编号、当前行号、当前字节号，将输出的总字节数，以及位于当前位置之前内容占整个文件的百分比。所有这些项目都参考位于输出的最后一行之后那一行的第一个字节。

v 调用编辑器，编辑当前正在检查的文件。编辑器名可通过环境变量 **EDITOR**、获得，或缺省为 **vi** 。如果 **EDITOR** 表示 **vi** 或 **ex** ，则会用选项调用该编辑器，使当前编辑器行是与调用时的 **more** 中当前位置相对应的物理行。

退出编辑器时， **more** 用当前行作为当前位置重写屏幕，来继续显示当前文件。

h 显示所有 **more** 命令的说明。

i/[**!**]*expression* 在文件中向前搜索包含正则 *expression* 的第 *i*- 行。 *i* 的缺省值为 1。搜索将从当前位置后的下一行开始。如果搜索成功，则会修改屏幕，以便使找到的行位于当前位置。空正则表达式 (**/Return**) 使用上一个正则表达式重复搜索。如果包括了字符 **!** ，则将搜索不包含 *expression* 的那些行。

如果 *expression* 的出现次数小于 *i* ，并且输入将为一个文件而非管道，则表达式在文件中的位置保持不变。

用户的 erase 和 kill 字符可用于编辑正则表达式。向后清除至第一列之前时会取消搜索命令。

*i***?**[**!**]*expression* 与 **/** 基本相同，但在文件中向后搜索包含正则 *expression* 的第 *i* 行。

注释：不同于编辑器， **?** 结构不应以 **/** 结尾。如果这样，则搜索模式会将末尾的斜线视为一个字符。

*i***n** 重复以前的搜索，即查找包含上一个 *expression* 的第 *i*- 行（缺省为 1），或者，如果以前的搜索为 **/!** 或 **?!** ，则搜索不包含上一个 *expression* 的行。

*i***N** 以先前搜索包含上一个 *expression* 的第 *i*- 行（缺省为 1）的相反方向重复搜索。

” （两个撇号）返回到上一次导致较大移动的命令的执行位置（“较大移动”定义为任何超过一整屏的移动）。如果未进行这种大的移动，则返回至文件开头。

!*command* 用 *command* 调用 Shell。 *command* 中的字符 **%** 和 **!** 分别被当前文件名和以前的 Shell 命令替代。如果没有当前文件名，则不会扩展 **%** 。序列 **\%** 和 **\!** 分别被 **%** 和 **!** 替代。

:e[*file*]
E[*file*] 检查一个新文件。如果未指定 *file* 参数，则会重新检查命令行上文件列表中的“当前”文件（请参阅 **:n** 和 **:p** 命令）。该文件名受 Shell 文字扩展过程的影响。如果 *file* 是 **#** （数字符号）字符，则会重新检查以前检查过的文件。

*i***:n** 检查下一个文件。如果指定了 *i* ，则检查命令行中指定的下一个第 *i*- 个文件。

*i***:p** 检查上一个文件。如果指定了 *i* ，则检查命令行中指定的上一个第 *i*- 个文件。

:t *tagstring* 转至提供的 *tagstring* ，并以该行作为当前位置滚动或重写屏幕。

m *letter* 用指定的字母标记当前位置，其中 *letter* 表示可移植字符集中的一个小写字母名。

’ *letter* 返回先前用 *letter* 标记的位置，使该行成为当前位置。

r
Ctrl-l 刷新屏幕。

R 刷新屏幕，忽略任何缓冲的输入。

. 点。重复先前执行的命令。

^ 停止文本的部分显示。 **more** 停止发送输出，并显示常规提示符。但有些输出会不幸丢失。

命令可立即生效；也就是说，不必按 **Return** 键。在输入命令字符本身之前，可用行终止字符取消正在形成的数字参数。

如果标准输出不是电传打字机，则 **more** 等效于 *cat*(1) 。

more 支持 **SIGWINCH** 信号，并根据窗口大小的变化重绘屏幕。

外部语言环境影响

环境变量

COLUMNS 覆盖系统选择的水平屏幕大小。

EDITOR **v** 命令用该变量来选择编辑器。

LANG 提供未设置或空国际化变量的缺省值。如果未设置 **LANG** 或为空，则使用缺省值“C”（请参阅 *lang*(5) ）。如果任一国际化变量中包含无效设置，则 **more** 就会认为所有国际化变量都设置为“C”。请参阅 *environ*(5) 。

LC_ALL 如果设为非空字符串值，则覆盖其他所有国际化变量的值。

LC_CTYPE 用于确定将文本解释为单字节和（或）多字节字符，将字符的类别解释为可输出，以及与正则表达式中的字符类表达式相匹配的字符的解释。

LC_MESSAGES
可确定应用于影响消息的格式和内容的语言环境，这种消息包括输出到标准错误的诊断消息，以及输出到标准输出的信息消息。

NLSPATH 用于确定消息清单的位置，以便处理 **LC_MESSAGES**。

LINES 覆盖系统选择的垂直屏幕大小，用作一整屏可显示的行数。在确定整屏可显示的行数时， **-n** 选项优于 **LINES** 变量。

MORE 用于确定包含选项、且选项前跟连字符并由空白字符分隔的字符串，正如命令行选项一样。所有命令行选项在 **MORE** 变量中的选项之后进行处理。在确定整屏可显示的行数时， **MORE** 变量优于 **TERM** 和 **LINES** 变量。

TERM 用于确定终端类型名。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

实际应用信息

当标准输出不是终端时，所有修改过滤器的选项都无效。这是基于以往的实践得出的结论。例如， **man** 的典型实现用管道将其输出传送至 **more -s** ，以便为终端用户压缩多余空白区域。但是 **man** 的输出被通过管道传送至 **lp** 时，则不希望进行这种压缩。

举例

要查看一个简单文件，可使用：

more *filename*

要预览 *nroff* 输出，可使用类似以下的命令：

nroff -mm +2 doc.n | more -s

如果文件包含表，可使用：

tbl *file* **| nroff -mm | col | more -s**

要在可显示 15 行的窗口中显示文件 **stuff** ，并将多个相邻的空白行转换为一个空白行，可使用：

more -s -n 15 *stuff*

要通过最后一整屏的内容检查每个文件，可使用：

more -p G *file1 file2*

要从第 100 行（当前位置 - 第三行，因此第 98 行为输出的第一行）开始检查每个文件，可使用：

more -p 100g *file1 file2*

要检查当前位置为第 30 行时包含 *tag* 标记字符串的文件，可使用：

more -t tag -p 30g

警告

标准错误，文件描述符 2 通常在交互使用时用于输入，且不应对其重定向（请参阅所使用的 Shell 联机帮助页的“输入/输出”一节）。

文件

/usr/share/lib/terminfo/?/* 编译的终端功能数据库

作者

more 由 Mark Nudleman、加州大学伯克利分校、OSF 和 HP 共同开发。

另请参阅

csh(1)、 man(1)、 pg(1)、 sh(1)、 term(4)、 terminfo(4)、 environ(5)、 lang(5)、 regexp(5)。

符合的标准

more： XPG4

名称

mpsched - 控制执行特定进程的处理器或位置域

概要

mpsched -h

mpsched -s

mpsched -g *command*

mpsched [**-P** *policy*] [**-f**] [**-T** *policy*] [**-l** *locality-domain-id*] [**-c** *spu*] *command*

mpsched [**-q**] [**-u**] [**-P** *policy*] [**-f**] [**-l** *locality-domain-id*] [**-c** *spu*] {**-p** *pid*}...

说明

mpsched 控制执行进程的处理器 (spu) 或位置域 (locality-domain-id)。通过将进程绑定到特定的处理器或位置域 (ldom) 或者设置进程的启动策略，可以实现该目的。

该命令可以用五种方式调用。

- 通过 **-h** ，它输出帮助消息。
- 通过 **-s** ，它返回系统硬件配置。其中包括有关系统中位置域编号和活动处理器的信息。
- 通过 **-g** ，它启用 *command* 及其 *arguments* 的成群调度。请参阅 *gang_sched*(7) 。
- 通过 **-P** 、 **-T** 或 **-l** 以及一个 *command* 及其 *arguments* ，它将绑定策略或启动策略应用于命令中。
- 通过 **-p** ，它将绑定策略或启动策略应用于指定的 *pid* 中。

选项

命令行选项包括：

-c *spu* 将指定的进程绑定到列出的 *spu* 。这将确保进程始终在指定的处理器上运行。在配置了处理器集 (pset) 的系统中，如果 *spu* 属于进程所绑定的同一个处理器集，则绑定将成功。

该选项可以与 **-P** 、 **-T** 和 **-p** 选项一起使用。

-f 如果系统中安装了 PRM，则允许将进程绑定到 *spu* 或 *ldom* 。

-g 对进程启用成群调度。不应将其他选项与 **-g** 选项一起使用。

-h 输出帮助消息。

-l *locality-domain-id*

将指定的进程绑定到列出的 *locality-domain* 。这将确保进程始终在指定的域中运行。在配置了处理器集 (pset) 的系统中，如果进程对应的 *ldom* 属于进程所绑定的同一个处理器集，则绑定将成功。

该选项可以与 **-P** 、 **-T** 和 **-p** 选项一起使用。

-p *pid* 指定进程 ID，即 *pid* 。要使用 **-p** 选项，调用者必须是拥有 **PRIV_MPCTL** 访问权限的组的成员，或是超级用户，或拥有与 *pid* 等效的用户 ID。指定命令而不是指定 **-p** 选项，则不需要特殊的权

限。虽然每个 **-p** 选项只能采用一个进程 ID，但每个命令行可以指定多个 **-p** 选项。

-q 向系统查询有关进程绑定的信息。这将返回进程是绑定到处理器还是位置域的相关信息。它还将报告有关进程的线程启动策略和进程启动策略的信息。如果该选项与 **-p** 选项一起使用，则只对指定进程进行查询。如果单独指定该选项，则系统中所有与缺省设置不同的进程的状态都将显示。

-s 输出系统硬件配置。不应指定其他选项。

-u 从现有的任何处理器或位置域绑定中解除进程的绑定。该选项只能与 **-p** 选项一起使用，并且不应指定其他选项。

-P *policy* 将指定的 *policy* 应用于进程。启动策略会影响从中产生进程的位置域。有关启动策略的详细信息，请参阅 *mpctl*(2) 联机帮助页。该选项可以与 **-T** 、 **-p** 、 **-c** 和 **-l** 选项一起使用。 *policy* 可以是下列值之一。

RR 循环启动策略。根据该策略，指定命令或进程的连续的直系子进程，以循环方式在与所创建进程相关的系统中的其他位置域中启动。

RR_TREE 基于树的循环启动策略。根据该策略，连续的子进程及其派生进程，以循环方式在与所创建进程相关的系统中的其他位置域中启动。

LL 最轻负载启动策略。根据该策略，子进程将于创建时在系统中负载最轻的位置域中启动。

FILL 填充第一个启动策略。根据该策略，指定命令或进程的连续的直系子进程将在与其父进程相同的位置域中启动，直到在该位置域中的每个处理器上都启动一个进程为止。这时，将在下一个位置域中创建新的进程。

FILL_TREE

基于树填充第一个启动策略。根据该策略，连续的子进程及其派生进程在与父进程相同的位置域中启动，直到在该位置域中的每个处理器上都启动一个进程为止。这时，将在下一个位置域中创建新的进程。

PACKED 压缩启动。根据该策略，连续的进程将在与其父进程相同的位置域中启动。永远不会选择其他域。

NONE 无特殊策略。将使用缺省的 HP-UX 启动策略。

-T *policy* 将指定的 *policy* 应用于进程的线程中。调度策略与 **-P** 选项的调度策略相同，不同的是，该调度策略应用于新创建的线程而不是进程。此外，只能在通过 **mpsched** 命令行启动的命令中指定线程策略。该选项可以与 **-P** 、 **-l** 和 **-c** 选项一起使用。

操作数

命令行操作数包括：

command 包括参数的命令。

返回值

如果成功调度 *command* ，则 **mpsched** 返回退出状态 0；如果调度失败，则返回退出状态 -1。

举例

在处理器 2 上执行 **a.out** 文件：

mpsched -c 2 a.out

将进程 ID 为 24217 的现有进程的进程启动策略设置为循环：

mpsched -P RR -p 24217

将进程 ID 为 1247 和 1842 的进程绑定到处理器 4：

mpsched -c 4 -p 1247 -p 1842

作者

mpsched 由 HP 开发。

另请参阅

getprivgrp(1)、 setprivgrp(1M)、 fork(2)、 getprivgrp(2)、 mpctl(2)、 privgrp(4)、 gang_sched(7)。

名称

mt - 磁带操作程序

概要

mt [**-f** *tapename*] *command* [*count*]

过时形式

mt [**-t** *tapename*] *command* [*count*]

说明

mt 用于将命令发送到磁带机。如果未指定 *tapename* ，则使用环境变量 **TAPE** ；如果未定义环境变量 **TAPE** ，则使用缺省驱动器。

mt 按要求的方向（向前或向后）绕带，当绕到指定 *count* 个 EOF 标记或记录后停止。如果未指定 *count* ，则假定一个计数。每个 EOF 标记均计数为一个记录。向后绕带时，磁带总是在 BOT 标记处停止，无论 *count* 中的数量是多少。

mt 接受下列 *command* ：

eof 写入 *count* 个 EOF 标记。

fsf 前进 *count* 个文件。

fsr 前进 *count* 条记录。

bsf 后退 *count* 个文件。

bsr 后退 *count* 条记录。

rew 倒带。

offl 倒带并且脱机

eod 搜索到数据结尾（仅适用于 DDS 和 QIC 驱动器）。

smk 写入 *count* 个设置标记（仅适用于 DDS 驱动器）。

fss 前进 *count* 个设置标记（仅适用于 DDS 驱动器）。

bss 后退 *count* 个设置标记（仅适用于 DDS 驱动器）。

status 输出关于磁带机的状态信息。

res 将磁带机专门留给发出 **mt** 命令的主机单独使用（仅适用于 stape 驱动程序）。

rel 将发出 **mt** 命令的主机对磁带机的单独使用解除（仅适用于 stape 驱动程序）。

走带操作（前进或后退文件或记录）使磁带定位到以运动方向走带的对象的更远处。即，后退一个文件使磁带位于文件标记之前，前进一个文件使磁带位于文件标记之后。这与磁带的所有标准用法一致。

警告

仅应指定原始的、非倒带的伯克利类型的设备。关闭时，此类型的设备不会重新定位磁带。这种设备的示例为 **/dev/rmt/0mnb** 。有关详细信息，请参考 *mt*(7) 。

在绕带的时候，有可能会超过 EOT 标记并绕过盘带的末尾。

只有发布了原始保留的主机释放了保留之后，才能清除保留。在存放保留的主机不可用的情况下，可以使用 **st** 命令，通过发布总线设备重置来回收设备。有关详细信息，请参阅 *st*(1M) 。

只能使用 *stape* 驱动程序，将保留/释放功能发布到驱动器。

举例

对与设备文件 **/dev/rmt/0mnb** 相关的磁带进行倒带:

```
mt -f /dev/rmt/0mnb rew
```

文件

/dev/rmt/*	原始磁带接口
/dev/rmt/0mnb	缺省磁带接口

作者

mt 由加州大学伯克利分校开发。

另请参阅

dd(1)、mt(7)、st(1M)。

名称

mv - 移动或重命名文件和目录

概要

mv [**-f**|**-i**] [**-e** *extarg*] *file1 new-file*

mv [**-f**|**-i**] [**-e** *extarg*] *file1* [*file2* ...] *dest-directory*

mv [**-f**|**-i**] [**-e** *extarg*] *directory1* [*directory2* ...] *dest-directory*

说明

mv 命令可以:

- 将一个文件 (file1) 移动到新文件或现有文件 (new-file)。
- 将一个或多个文件 (file1, *[* file2, *...]*) 移动到现有目录 (dest-directory)。
- 将一个或多个目录子树 (directory1, *[* directory2, *...]*) 移动到新目录或现有目录 (dest-directory)。

将 *file1* 移动到 *new-file* 可用于重命名目录内的文件，或者，在某个文件系统内或不同文件系统中重定位文件。当目标为目录时，会将一个或多个文件移动到该目录。如果要移动两个或多个文件，则目标必须是一个目录。在将单个文件移动到新文件时，如果存在 *new-file* ，则会删除其内容。

如果 *dest-directory* 或现有目标文件 *new-file* 的访问权限禁止写入，则 **mv** 会要求覆盖文件的权限。其实现过程是：输出模式（请参阅 *chmod*(2) 和下面的“访问控制列表”），后跟当前语言环境的语言中 *yes* 和 *no* 的首字母，提示您作出响应，然后从标准输入读取一行。如果响应是肯定的且操作是允许的，则将进行该操作；否则，该命令会继续处理下一个源文件（如果有的话）。

如果 *file1* 是一个文件，而 *new-file* 是指向具有其他链接的另一文件的链接，则其他链接保持不变，而且 *new-file* 变成新文件。如果 *file1* 是具有链接的文件或指向文件的链接，则现有文件或链接将保持不变，但名称会更改为 *new-file* （它可能位于也可能不位于 *file1* 所在的目录中，具体取决于在 **mv** 命令中使用的目录路径名）。被移动的文件的上次访问和修改时间保持不变。

选项

mv 可识别下列选项:

-f 执行 **mv** 命令，但不提示权限。当标准输入不是终端时，使用此选项。

-i 使 **mv** 在移动将覆盖现有文件的文件之前，将一条提示写入到标准输出。如果来自标准输入的响应是肯定的，则在权限允许移动的情况下将会移动文件。

-e *extarg* 指定如何处理要移动的文件的任何盘区属性。 *extarg* 可以是以下值之一:

- **warn** 如果无法保留盘区属性，则将发出一条警告消息，但是仍将会移动文件。
- **ignore** 不保留盘区属性。
- **force** 如果无法保留盘区属性，则不移动文件。

如果与单个目标目录一起指定了多个源文件，则 **mv** 将移动没有盘区属性或者具有可保留的盘区属性的文件。如果文件无法保留其盘区属性，则 **mv** 将不移动它。

如果要将文件移动到不支持盘区属性的文件系统，或者该文件系统的块大小与原始块大小不同，则无法保留盘区属性。如果未指定 **-e** ，则 *extarg* 的缺省值为 **warn** 。

访问控制列表 (ACL)

如果可选的 ACL 条目与 *new-file* 关联，则在要求覆盖文件的权限时， **mv** 会在访问模式之后显示一个加号 **(+)** 。

如果 *new-file* 是一个新文件，则它继承 *file1* 的访问控制列表，并加以更改来反映这两个文件之间的任何所有权差异（请参阅 *acl*(5) 和 *aclv(5)* ）。在 JFS 文件系统中， **mv** 创建的新文件不继承其父目录的缺省 ACL 条目（如果有的话），而是保留其原始 ACL。在将文件从 JFS 文件系统移动到 HFS 文件系统（或反向移动）时，会丢失可选的 ACL 条目。

外部语言环境影响

环境变量

LC_CTYPE 用于确定将文本解释为单字节字符和（或）多字节字符。

LANG 和 **LC_CTYPE** 可确定 y 的本地语言等效项（对于“是/否”查询）。

LANG 用于确定显示消息的语言。

如果未在环境中指定 **LC_CTYPE** 或将其设置为空字符串，则 **LANG** 的值会用作每个未指定变量或空变量的缺省值。如果不指定 **LANG** 或将其设置为空字符串，则使用缺省值 **C** （请参阅 *lang*(5) ）而非 **LANG** 。如果任一国际化变量包含无效设置，则 **mv** 就会认为所有国际化变量都设置为 **C** 。请参阅 *environ*(5) 。

国际代码集支持

支持单字节字符代码集和多字节字符代码集。

举例

重命名当前目录中的文件：

mv old-filename new-filename

重命名当前目录中的目录：

mv old-dirname new-dirname

重命名当前目录中的一个文件，该文件的名称以非打印控制字符或 Shell 专用的字符（如 **-** 和 ***** ）开头（根据情况的不同，可能需要特别小心）：

mv ./bad-filename new-filename
mv ./?bad-filename new-filename
mv ./*bad-filename new-filename

将目录 **sourcedir** 及其内容移动到文件系统中的新位置 (**targetdir**) ，完成后，名为 **sourcedir** 的子目录将位于目录 **targetdir** 中：

mv sourcedir targetdir

将当前目录中的所有文件和目录（包括链接）移动到 **targetdir** 下的新位置：

mv * targetdir

将 **sourcedir** 中的所有文件和目录（包括链接）移动到 **targetdir** 下的新位置 （**sourcedir** 和 **targetdir** 位于不同的目录路径中）：

mv sourcedir/* targetdir

警告

如果 *file1* 和 *new-file* 存在于不同的文件系统上，则 **mv** 会复制该文件并删除原始文件。在这种情况下，移动方将成为所有者，并且将丢失与其他文件的任何链接关系。 **mv** 无法跨文件系统移动硬链接。如果 *file1* 是目录，则 **mv** 会将整个目录结构复制到目标文件系统并删除原始目录。

mv 不能用于执行以下操作：

- 使用 **.** 或 **..** 符号重命名当前工作目录或其父目录。
- 将目录重命名为与同一父目录中包含的某一文件完全同名的新名称。

相关内容

NFS

汇总联网文件的访问控制列表（如 *stat*(2) 在 *st_mode* 中返回的那样），但是不复制到新文件。当对这样的文件使用 **mv** 时，如果要求覆盖文件的权限，则在模式值之后不输出 **+** 。

作者

mv 由 AT&T、加州大学伯克利分校和 HP 开发。

另请参阅

cp(1)、 cpio(1)、 ln(1)、 rm(1)、 link(1M)、 lstat(2)、 readlink(2)、 stat(2)、 symlink(2)、 symlink(4)、 acl(5)、 aclv(5)。

符合的标准

mv：SVID2、SVID3、XPG2、XPG3、XPG4、POSIX.2